TURBO news

Revista para Computadores ATARI®

Nº 4 - Octubre 1989

$ 550

Comunicaciones Vía Modem

DESPACHO A REGIONES I - II - XI y XII S 40

ADQUIERELOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB KW VIDEO LA ESPANOLA • **VIÑA DEL MAR:** FALABELLA VINA INSIS MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC CASA ROYAL CENTRO ATARI COMERCIAL ESTADO COMPUMANQUE COMPUCENTER FALABELLA AHUMADA FALABELLA P. ARAUCO IMACO INFOGROUP PC STORE PETERSEN ROLEC SUPERMERCADOS UNIMARC TASCO VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUNIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" MULTICENTRO VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB DISMAR DISMAR 2 EQUS PHANTER RAPSODIA SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE ESTABLECIMIENTOS GEJMAN FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE DIMARSA • **COYHAIQUE:** FACI HOGAR • **PUNTA ARENAS:** BALFER LTDA.

CONTENIDO

El mundo de las Finanzas al alcance de todos	2
Atari por dentro Antic y el movimiento fino	5
Turbo Mail	6
Curso de Basic (Lección 4)	8
Desarrollando Hardware Intercomunicando computadores	12
Mapa de memoria	15
Atari en línea	17
Curso de Assembler (Lección 4)	21
Ranking del mes Descripción de juegos	26
Programas	28

TURBO news

Circulación mensual. Nacional e Internacional
Destinada a los usuarios de computadores ATARI (R) como material didáctico de PROGRAMACION TURBO news (R) es una publicación de EDITORA TURBO LTDA.
Domicilio Av. Holanda 2456 - Teléfono: 2238063
SANTIAGO CHILE.

GERENTE: Arturo Valdivieso Martínez. **EDITOR:** Editora Turbo Limitada. **PRODUCCION:** Pedro P. Caraball Alvarez - Marcelo A. Waldbaum Olszevicki - Mauro Pieressa Schachtel, Programadores y Diseñadores de Computación **COLABORACION:** Silvia Edelstein. **PERIODISTA:** Rodrigo Manríquez. **ARTE/DISEÑO:** Odalí Guerrero López. **FOTOGRAFIAS:** Isabel Bellalta.

DEPARTAMENTO DE VENTAS Y PUBLICIDAD: Arturo Valdivieso M. Hernán Vittini G. Agradecemos la colaboración de: Coelsa S.A. Centro Atari (Augusto Leguía Sur 75). ATARI, es marca registrada de Atari Corporation. TURBO, es marca registrada de EDITORA TURBO LIMITADA (Reg. Marc. Nº342428 9-05-89) Impresa en los talleres de Impresos Nova Ltda. quien actúa como Impresora.

EDITORIAL

Después de este largo mes de trabajo nos reconforta encontrarnos con ustedes nuevamente.

Esto demuestra que todos necesitamos comunicarnos. Para vivir, las personas deben intercambiar ideas de todo tipo, pueden ser complejos conceptos o sentimientos cotidianos.

Cuando queremos saber algo, para trabajar, estudiar, divertirnos o simplemente sentirnos mejor tras la actividad de un largo día, recurrimos a una fuente de información. De allí obtenemos datos que enriquecen nuestra vida.

Tras recibir esa información, podemos enviar un mensaje de respuesta. Así cerramos un círculo de comunicación.

Ese es el tema principal que abordamos en este número.

Los computadores ya demostraron su idoneidad en el almacenamiento y proceso de la información.

Alguien pensó cómo comunicarlos entre sí. Y así comenzó la historia del Modem, un novedoso aparato que comienza a difundirse en Chile.

Hemos dedicado una nueva sección de esta revista a las comunicaciones vía Modem. Esperamos con ellos satisfacer tus inquietudes al respecto y como siempre, mantenerte informado de todas las novedades vinculadas con tu compañero ATARI.

COMPUTADOR DESDE LA CASA AL BANCO ESPAÑOL-CHILE

EL MUNDO DE LAS FINANZAS AL ALCANCE DE TODOS

Desde hace algunos años, el traspaso de información comenzó a ser más rápido. En Europa y Estados Unidos, los datos viajaban con una velocidad casi instantánea.

Nuestro país no quedó ajeno al cambio.

Frente a esta nueva situación, y para lograr que desde el hogar se puedan manejar las actividades financieras familiares, es que el Banco Español Chile inició un servicio que permite una intercomunicación inmediata.

Con el uso del Modem es posible a través de la línea telefónica obtener informaciones en la pantalla de nuestro televisor

Es el "Banco en Casa por Computador"

Para usar este nuevo servicio se necesita un computador Atari de ocho bits. Este puede ser un modelo 800 XL, un 65 XE o un 130 XE.

Otro elemento fundamental para traspasar la información es el Modem. Este aparato convierte los impulsos del ordenador en señales que pueden ser transmitidas por una simple línea telefónica.

Una vez contratado el servicio, se puede conectar el computador al Modem y éste a la roseta del teléfono. Así, en lugar de discar, basta anotar el número telefónico del banco en la pantalla de nuestro televisor. Desde el otro extremo de la línea aparece la información necesaria sobre nuestros movimientos bancarios.

De esta forma, el computador de nuestra casa está vinculado al Modem del banco. Desde allí la información pasa a un conversor de protocolo que selecciona los mensajes aceptados por el computador central del banco.

Para ingresar al sistema existe un programa que el Banco Español Chile entrega dentro de un cartridge. Este permite entrar a la cuenta del usuario sólo después de digitar un número secreto que conoce únicamente ese cliente del banco.

Así, este cliente estudia sus cartolas con los últimos movimientos de cuenta corriente y si tiene una impresora, puede imprimirlos para guardarlos. Además existe la posibilidad de revisar los movimientos de cuenta a partir de una fecha deseada.

Otra posibilidad es obtener el saldo en línea de una cuenta corriente, en pesos o dólares, resumido o detallado. También se puede averiguar el monto de un cheque que no fue anotado en la Chequera o pedir la emisión de la cartola.

El nuevo servicio permite cotizar el valor de las principales monedas extranjeras y consultar la cuenta de cobranza de letras.

Según Fernando Casasbellas, Gerente de Productos del Banco Español, se puede consultar, a través de la pantalla, el pago de las cuentas de agua, electricidad y teléfono, y próximamente se incorporará una operación que permitirá pagar desde la pantalla.

Un servicio adicional que el

sistema ofrece es el Correo Electrónico. Con él, desde la casa, se pueden enviar y recibir mensajes para solicitar operaciones financieras.

Todos estos servicios están incluidos en un Menú que aparece en el televisor. De ellas se seleccionan sólo tres en cada llamada telefónica. Esta medida evita una saturación del sistema por parte de algún cliente que efectúe consultas durante un tiempo muy prolongado.

Casasbellas explicó que la nueva forma de comunicación permite dar un uso útil al computador casero, cuya actividad habitualmente está restringida a los juegos de los niños. El proyecto tiene como potenciales usuarios a los dueños de más de 100 mil computadores Atari que hay en Chile.

Además, se eligió esa marca pues su lenguaje es más comprensible para el uso familiar.

El Modem casero apropiado para entrar a este sistema es el TH 003, de la norma CCITT, con capacidad para 300 baudios. Este será recibido en el banco por uno automático de 300 a 2400 baudios.

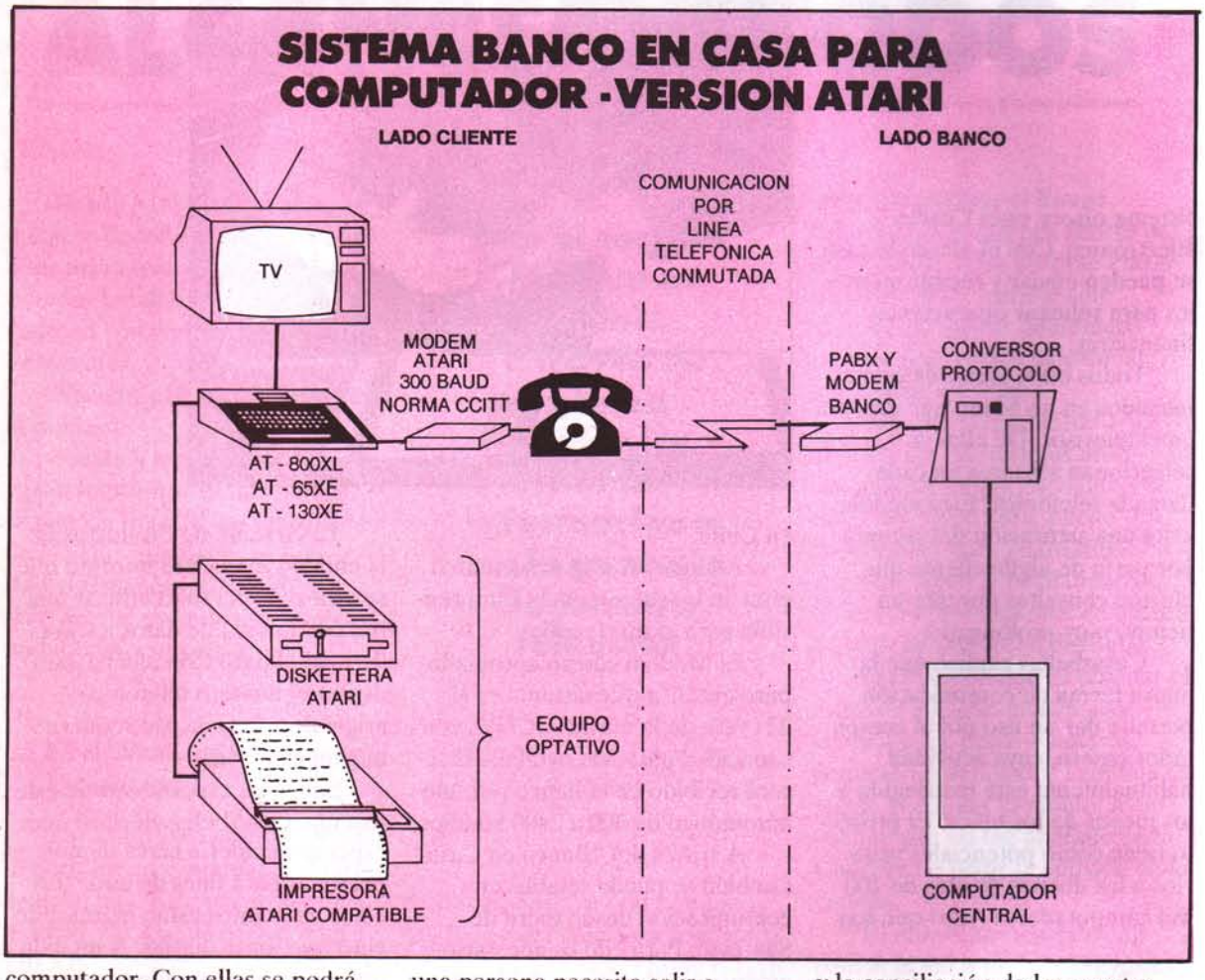
A través del "Banco en Casa" también se puede establecer comunicación desde fuera de Santiago. Para ello es necesario anteponer, al número del banco, el código telefónico correspondiente a Santiago (02).

El Gerente de Productos de la entidad financiera informó que se está considerando utilizar una red conmutada de datos a través de todo el país. Con ella bastará digitar el número telefónico asignado a la zona, sin requerir ningún otro código.

Sobre las expectativas de este sistema, Casasbellas declaró que esperan atender a cerca de mil personas para fines de año. "En estos momentos están ingresando cinco personas diarias. A medida que sigan aumentando vamos a instalar mayor número de líneas", manifestó el ejecutivo.

El representante del Banco Español Chile agregó que en breve plazo se aumentarán las operaciones posibles a través del

EL MUNDO DE LAS FINANZAS AL ALCANCE DE TODOS



computador. Con ellas se podrá
- Pagar cuentas de servicios
- Solicitar créditos y consultar por una operación urgente
- Transferir fondos entre cuentas
- Consultar detalles de cupos y deuda tarjeta VISA
- Dar orden de bloqueo de Tarjeta de Crédito

"Lo único que no se podrá hacer es sacar plata, pero para eso están los cajeros automáticos. Si una persona necesita salir a comprar algo, podrá averiguar antes si tiene dinero en su cuenta", destacó Casasbellas.

Otra posibilidad que entrega el computador casero conectado al Banco es una aplicación financiera. Si se cuenta con una disquetera XFF 551 se podrá usar el disco con programas que entrega el mismo Banco. Estos permiten el cálculo del presupuesto familiar y la conciliación de las cuentas corrientes. También se pueden realizar operaciones financieras como el cálculo de créditos.

Finalmente, Casasbellas expresó que Chile es uno de los pocos países donde se ha desarrollado un sistema como el "Banco en Casa" con un computador de uso masivo.

ANTIC Y EL MOVIMIENTO FINO

ATARI POR DENTRO

por: PEDRO P. CARABALL A.

Para terminar con ANTIC vamos a estudiar una importante posibilidad que nos permite este procesador.

Los movimientos finos Vertical y Horizontal constituyen una elegante manera de presentar un título, un menú, el o los autores de un determinado programa,etc.

ANTIC permite manejar estos movimientos con mucha facilidad para el programador en Assembler, como podrán corroborar en el listado que se encuentra en las páginas correspondientes.

Programadores BASIC: A no desesperar! Se incluye también un listado en BASIC que te permite crear un string de 75 caracteres y mostrarlo permanentemente en la pantalla con un movimiento fino horizontal en una línea que se encuentra más arriba del modo gráfico que estés usando.

El programa que se incluye reside permanentemente en la página 6. Esto quiere decir que no puedes disponer de las posiciones de memoria que se encuentran entre 1536 y 1791 ($600 a $6FF) para otros usos.

```
MOVIENTO FINO
100 REM ***************************
110 REM *     REVISTA TURBO NEWS  *
120 REM *       ATARI POR DENTRO  *
130 REM *   EJEMPLO DE MOVIMIENTO *
140 REM *   FINO CON INTERRUPCION *
150 REM *   DE BLANQUEO VERTICAL  *
160 REM *                         *
170 REM *     PEDRO P. CARABALL A.*
180 REM ***************************
190 REM
200 RESTORE :C=0
210 FOR I=1536 TO 1673
220 READ B:C=C+B
230 POKE I,B
240 NEXT I
250 IF C<>14965 THEN ? "ERROR DE DATA"
:END
260 FOR I=1694 TO 1769:POKE I,0:NEXT I

270 POKE 710,0:DIM A$(75)
280 ? "INGRESA EL TEXTO A DESPLEGAR"
290 ? "  (MAXIMO 75 CARACTERES)"
300 ? ">";:INPUT #16;A$
```

Aunque este movimiento es muy fácil de implementar, en caso de hacerlo en BASIC nos encontramos con un problema de velocidad, que hace imposible un suave y constante deslizamiento del texto. Por ello nos vemos obligados a usar una rutina de interrupción en el blanqueo vertical o con cualquier otra base de tiempo disponible.

Si tienes alguna duda respecto a ANTIC que no haya sido vista en estas páginas, no dejes de escribirnos y tendrás una respuesta aclarando tus dificultades.

Esperamos que con el dominio de ANTIC que habrás conseguido a esta altura del camino, puedas hacer que tus programas tengan una magnífica presentación, la cual deseamos te dé muchos beneficios.

En nuestros próximos números seguiremos incursionando en este increíble mundo de ATARI, poniendo al alcance de tu mano la más completa y mejor información.

TURBO MAIL

Sres. "TURBO news":

Junto con felicitarlos por su excelente revista y darles todo mi apoyo y adhesión como hermanos Atarianos, respondo a la sección "TURBO MAIL", enviando en estas líneas mis dudas, datos personales y sugerencias.

Después de releer varias veces la revista, me surgieron una serie de preguntas como éstas:

- Una cassettera normal, necesita modificación alguna para utilizar el TURBO TENIS. Cómo adquirirlo y en qué precio se vende en el mercado?.

- Cómo tengo que hacer para subscribirme a TURBO news, si no gano el sorteo?.

- Se pueden mandar colaboraciones sólo impresas para cuidar el bolsillo?.

Mis datos personales son los siguientes:

Nombre: Conrado Pacheco A.
Edad: 17 años.
Domicilio: Bartolo Soto 3861 - San Miguel.
Configuración: Atari 800 XL y cassettera XC-12.
Experiencia en Atari: 3 años como programador autodidacta y 7 meses como profesor particular de Atari BASIC.

En realidad, quiero darles algunas sugerencias para que tengan en cuenta la opinión de sus lectores:

- En la sección Turbo Cosas, describan y enseñen a utilizar el mapa de memoria ATARI, así como también los PEEK y POKE.

- En el curso de Assembler, publiquen una guía en castellano de todos las instrucciones del lenguaje.

- No publiquen las rutinas en formato STRING, ya que la simbología ATASCII es poco legible y se presta a confusión cuando tipeamos sus programas.

Quisiera también compartir con Uds. mis ansias de que llegue el próximo mes, para ver en qué sigue el curso de Assembler y el desarrollo de Hardware, los cuales fueron los temas que más me tocaron, ya que después de 3 largos años de ver y trabajar los distintos aspectos del 800 XL, uno ya le va encontrando trabas al BASIC y eso imposibilita al programador medio alcanzar y dominar las capacidades de este "pequeño gigante".

Afectuosamente.
Conrado Pacheco A.

MANTENTE EN CONTACTO

Conrado:

En primer lugar te agradecemos las expresiones de elogio hacia nuestra Revista. En ella volcamos el conocimiento y la experiencia que hemos adquirido durante varios años de trabajo con el Atari. Nuestro objetivo no es otro que compartir nuestra información con todos los fanáticos de este computador y hallamos en TURBO news el medio adecuado para hacerlo.

Con respecto a tu pregunta sobre el Sistema de Carga de Cassettes Turbo Tenis, te aclaramos que no es necesaria ninguna modificación en tu Unidad de Cinta. El Turbo Tenis está preparado para funcionar con cualquier cassettera compatible con el computador Atari. En nuestra Revista, como podrás apreciar se encuentra una publicidad de Turbo Software, marca que tiene incorporada en todos sus cassettes (de juegos, educativos y los del Light Gun Atari), este sistema de carga. Es decir, que podrás adquirir los cassettes con el sistema TURBO TENIS en los distribuidores que se encuentran mencionados en dicha promoción.

Para suscribirte a nuestra revista, sólo tienes que completar el Cupón de Suscripción correspondiente.

En referencia a tu inquietud por las colaboraciones, hemos pensado la alternativa de estudiar el material del disco o cassette que tú nos envíes y luego devolvértelo. Si recibiéramos las colaboraciones en papel, tendríamos que tipearlas para analizar el contenido y en tiempo sería prácticamente imposible hacerlo, en virtud de la cantidad que recibimos.

Como habrás notado, en nuestra anterior edición de TURBO news, incorporamos una sección especial, que trata el Mapa de Memoria de tu Atari y la forma en que debes utilizar las instrucciones PEEK y POKE para sacar mejor provecho de tu computador.

Te agradezco en forma particular tu sugerencia respecto al curso de Assembler. Nos pareció tan acertada que la implementamos en esta seccion del curso.

En cuanto a las rutinas en formato ATASCII, procuraremos de ahora en más evitar su uso.

Nos despedimos entonces, deseando que sigas en contacto con nosotros, enviando tus cartas y colaboraciones que nos dan ánimos para seguir adelante.

Marcelo Waldbaum

Recibimos en nuestra Editorial una simpática carta del joven lector Marco Antonio Burgos Reyes quien nos envía un bonito dibujo de uno de los juegos de la línea Turbo Software, el MOON PATROL. Queremos en este espacio agradecer tu gesto e informarte que puedes pasar por nuestra Revista a retirar otro juego de la línea TURBO SOFTWARE que has ganado por tu hermosa colaboración.

MARCELO WALDBAUM

Continuando con el curso de Basic, daremos en este número las instrucciones necesarias para poder trabajar con una impresora, con variables alfanuméricas y otros trucos muy útiles para utilizar en programación. También presentamos otro programa explicado. Se trata de un programa muy sencillo de facturación, para que puedas practicar todo lo dado hasta el momento.

BA

Hasta ahora, en los números anteriores, hemos trabajado únicamente con variables numéricas, es decir aquellas que sólo pueden contener números. Pero en la práctica,se hace imprescindible contar también con variables alfanuméricas, o sea aquellas que pueden almacenar letras o números. Cabe destacar que aunque una variable alfanumérica posea solo números no se pueden realizar operaciones matemáticas entre ellas. Ya hemos visto que la forma que tiene el computador de reconocer una variable alfanumérica, se debe a que sus nombres finalizan con el símbolo $. Por ejemplo: NOMBRE$, LOCALIDAD$, etc. Las variables numéricas, en cambio, necesitan "declararse" al principio de cada programa, indicando allí cuál es su longitud. Esto se logra con la instrucción DIM

INSTRUCCION DIM

Su significado es "dimensión". Se utiliza para indicar la longitud de las variables alfanuméricas. Pueden dimensionarse varias variables con una sola instrucción DIM, separándolas entre sí por el símbolo ",". También es utilizada para dimensionar vectores y matrices pero este tema lo veremos en el futuro. Ejemplo:

```
DIM NOMBRE $(20),
DOMICILIO$(30)
```

Después de ejecutarse esta instrucción, quedan listas para ser utilizadas las variables NOMBRE$ y DOMICILIO$, admitiendo cada una, una longitud máxima de 20 y 30 caracteres respectivamente. Esas longitudes pueden ser tan grandes como uno desee, siempre cuidando de no exceder la capacidad de memoria del computador.

Existen maneras de referirse a una parte de una variable alfanumérica. Por ejemplo, si deseamos averiguar con qué letra comienza un nombre. Esto se logra encerrando entre paréntesis la posición inicial y final que queremos considerar. En el ejemplo mencionado, para imprimir la primera letra de una variable, se debe hacer:

```
PRINT NOMBRE$(1,1)
```

Si no se pone la segunda posición, asume que es hasta el final. Por ejemplo:

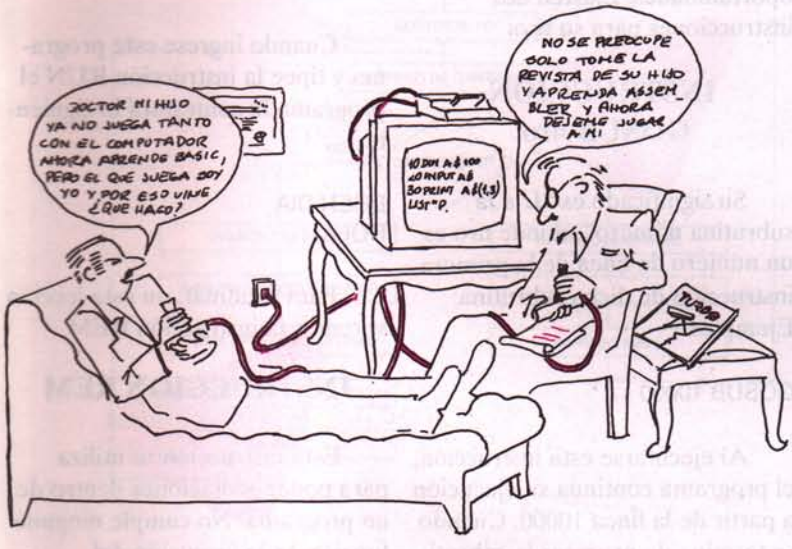
```
PRINT NOMBRE$(2)
```

Imprime todo el nombre excepto la primer letra.

Hemos visto ya, en números

SIC

LECCION 4



anteriores, que no pueden mezclarse en una asignación, por ejemplo, una variable numérica con una alfanumérica. Sería el caso de hacer:

NUMERO = EDAD$

Da error. Para ello existen las instrucciones VAL y STR$ que convierten una variable alfanumérica en numérica y viceversa respectivamente.

INSTRUCCION VAL

Su significado es "valor". Su sintaxis es la siguiente.

EDAD = VAL(EDAD$)

La variable entre paréntesis debe ser alfanumérica. Con la variable EDAD sí se pueden realizar operaciones matemáticas.

INSTRUCCION STR$

Su significado es "STRING" y en castellano sería "texto". Su sintaxis es la siguiente:

NUMERO$ = STR$(NUMERO)

La variable entre paréntesis debe ser numérica. En el ejemplo que acompaña este artículo se ve una de las aplicaciones que puede tener esta instrucción.

Otra instrucción que existe para el manejo de variables alfanuméricas, es la utilizada para averiguar cuántos caracteres del total de la variable han sido ocupados. La misma es LEN$.

INSTRUCCION LEN$

Su significado es "LENGTH" que en castellano significa "LONGITUD". Su sintaxis es la siguiente:

PRINT LEN$(NOMBRE$)

El computador contestará con la cantidad de caracteres que se han utilizado del total de la variable NOMBRE$.

Otra diferencia que existe entre las variables numéricas y alfanuméricas, es que cuando a una numérica se le asigna un nuevo contenido, si anteriormente la misma poseía algún valor este se borra automáticamente. En el caso de las variables alfanuméricas esto no sucede. Por ello, si no se borra la variable, puede quedar basura dentro de la misma. Por ejemplo, si se hace:

NOMBRE$ = "MARCELO"
NOMBRE$ = "MAURO"
PRINT NOMBRE$

El computador va a imprimir:

MAUROLO

Existen varias maneras de borrar la variable. La forma más rápida es ejecutar la siguiente serie de instrucciones. Para el ejemplo, vamos a suponer la existencia de la variable NOM$ de longitud 40. Para otras longitudes

BASIC

bastará con cambiar el número 40 por el que corresponda.

```
NOM$(1)=" "
NOM$(40)=" "
NOM$(2)=NOM$
```

El proceso de borrado se hace automáticamente debido al sistema que tiene el computador para efectuar una asignación. Esto es absolutamente invisible para el usuario.

A continuación comenzaremos a ver las instrucciones existentes para el uso de la impresora.

INSTRUCCION LPRINT

Es el equivalente de la instrucción PRINT, pero para la impresora. Todo lo explicado anteriormente para aquella, es válido para esta. Ejemplo:

```
LPRINT NOMBRE$
```

Después de ejecutada esta instrucción, se imprimirá el contenido de la variable NOMBRE$.

Otra instrucción para utilizar con la impresora es: LIST "P:" para la impresora. Se utiliza para listar los programas que hayas ingresado en tu computador.

INSTRUCCION LIST "P:"

Esta instrucción es equivalente a la instrucción LIST descrita en los números anteriores pero

Muchas veces, en la realización de un programa vemos que hay partes del mismo que son iguales entre sí, produciéndose un desperdicio de memoria y pérdida de tiempo para el programador. Para solucionar este problema se han creado las Subrutinas. Estas son porciones de programa que van a ser utilizadas en varias oportunidades. Existen dos instrucciones para su uso:

INSTRUCCION GOSUB nro.

Su significado es "Ir a la subrutina número", donde nro es un número de línea de la primera instrucción de dicha subrutina. Ejemplo:

```
GOSUB 10000
```

Al ejecutarse esta instrucción, el programa continúa su ejecución a partir de la línea 10000. Cuando se termina de procesar la subrutina, debe proseguir a continuación de la instrucción desde donde fue llamada. Esto se logra con la instrucción RETURN.

INSTRUCCION RETURN

Su significado es "volver". Cuando el programa encuentra este comando, el flujo del mismo continúa en la instrucción que sigue a la última instrucción GOSUB. Ejemplo:

```
100 GOSUB 1000
120 END
1000 PRINT "BUEN DIA"
1010 RETURN
```

Cuando ingrese este programa y tipee la instrucción RUN el programa le contestará lo siguiente:

```
BUEN DIA
HOLA
```

Para terminar, en esta lección veremos la instrucción REM.

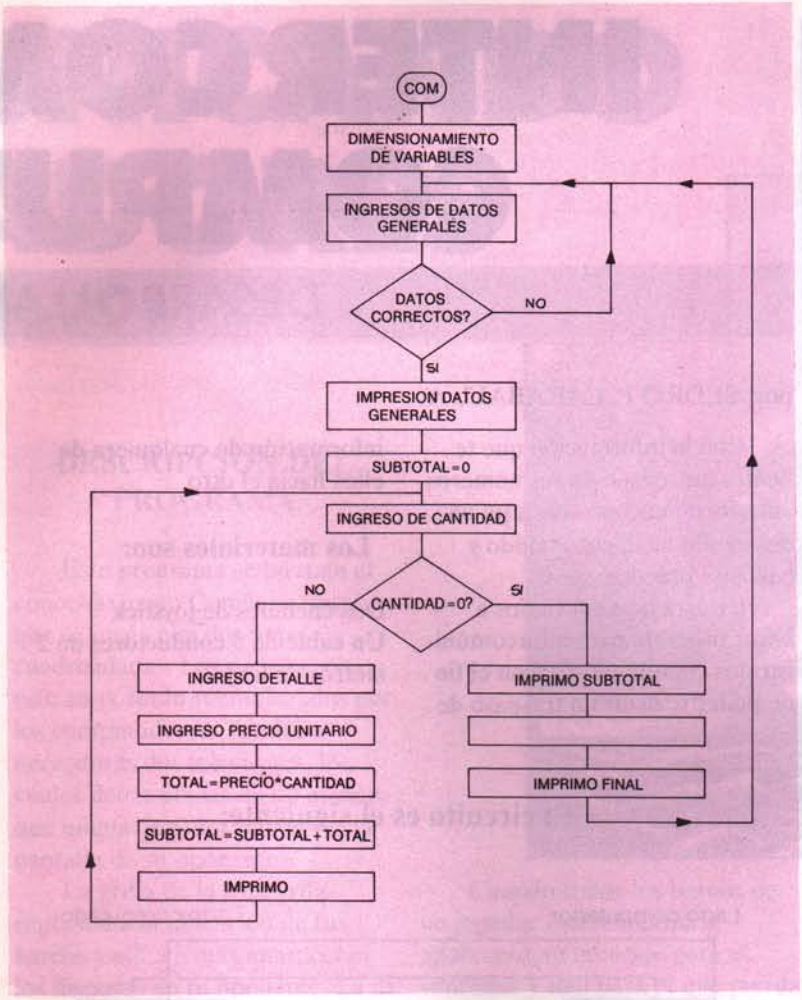
INSTRUCCION REM

Esta instrucción se utiliza para poner acotaciones dentro de un programa. No cumple ninguna función en la ejecución del mismo. Sirve exclusivamente para agregar claridad al programa.

A continuación veremos la aplicación de todas estas instrucciones en un programa de facturación bastante sencillo.

Comenzaremos como siempre con el diagrama de flujo:

BA



Recuerda que para detener el programa deberás oprimir la tecla BREAK.
Para ver el listado de este programa deberás referirte a la parte correspondiente al final de la revista.

Comencemos a analizarlo. En las líneas 70 y 80 dimensionamos todas las variables que vamos a utilizar. Las longitudes que fueron definidas pueden ser modificadas a criterio del lector.

En la línea 90 blanqueamos la variable LINEA$. Esta variable va a ser utilizada para armar cada línea de impresión. Podría haberse obviado e imprimirla como veremos más adelante.

De la línea 100 a la 170 ingresamos los datos generales del cuerpo de la factura. A cada línea le agregamos una instrucción TRAP para cuidar de no excedernos en la longitud que le hemos asignado a cada variable. Nota que al teléfono podríamos haberlo puesto en una variable numérica, sin embargo es recomendable que si no vamos a hacer operaciones matemáticas con dichas variables, se utilicen las alfanuméricas. Con ellas existen menos posibilidades de cometer errores, porque aceptan indistintamente letras y números.

En la línea 180 verificamos que todos los datos ingresados hayan sido correctos, ya que en caso contrario deberán ingresarse nuevamente. En las líneas 200 a 230 preparamos las líneas de impresión para posteriormente imprimirlas.

Como dijimos anteriormente no era necesario armar la línea de impresión. Por ejemplo, la línea 210 puede reemplazarse por la siguiente:

```
LPRINT NOMBRE$,,,RUT$
```

La cantidad de "," en este ejemplo es aproximado. Se dificulta hacer que las variables caigan en alguna columna exacta. En el caso del programa actual, utilizando la variable LINEA$, podemos poner cualquier variable en su posición correcta muy fácilmente.

BASIC

De la línea 300 a 360 ingresamos cada una de las líneas de detalle. Para concluir el ingreso de las líneas de detalle deberá ingresarse el numero 0. En la variable TOTAL guardamos el resultado de multiplicar el precio unitario por la cantidad de cada uno de los ítems. En la variable SUBTOTAL, vamos acumulando todos los precios totales de los ítems. En la línea 350 armamos la línea correspondiente a cada detalle y la imprimimos.

Cuando ponemos un 0 en la cantidad imprimimos el SUBTOTAL, el 16% del IVA y el total IVA incluido.

La rutina que se encuentra a partir de la línea 1000 es la que imprime la línea y luego la limpia para esperar la siguiente.

Si no posees impresora, igual te conviene estudiar el programa, para afianzar tus conocimientos.Te retamos a que lo modifiques para que salga por la pantalla. Debes saber que la misma tiene 40 columnas en lugar de las 80 de la impresora, por lo que,además de reemplazar las variables LPRINT por PRINT, deberás tener cuidado al armar las líneas, para no excederte de las 40 columnas.

En los próximos números continuaremos interiorizándonos en el fascinante mundo de la computación. HASTA PRONTO!!!

MAURO PIERESSA S.

INTERCOM
COMPUT

DESARROLLAND

por: PEDRO P. CARABALL A.

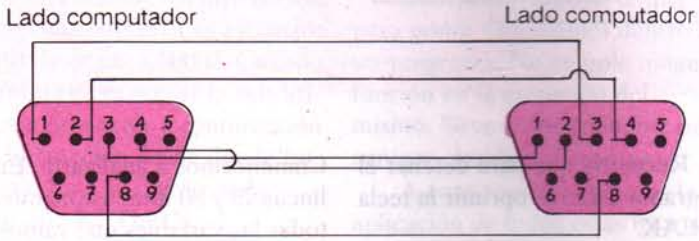
Con la información que te hemos entregado en los números anteriores, encontraras aquí un desarrollo fácil, entretenido y bastante práctico.

En esta ocasión vamos a hacer un cable para intercomunicar dos computadores, con el fin de poder realizar un traspaso de información de cualquiera de ellos hacia el otro.

Los materiales son:

Dos enchufes de joystick
Un cable de 5 conductores de 2 metros de largo.

El circuito es el siguiente:



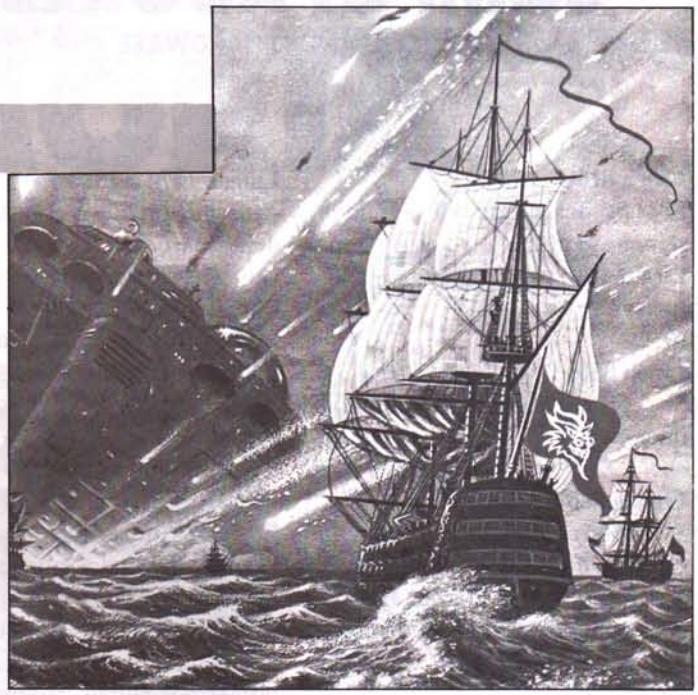
Una vez que tengas armado tu cable de interconexión, conéctalo en el enchufe del joystick B (posterior) de ambos computadores y estarás listo para recibir o transmitir datos.

En la sección de programas encontrarás el programa COMBATE NAVAL, el cual te permitirá utilizar el cable que acabas de armar para jugar este entretenido juego con un amigo que tenga un computador ATARI. Además podrás usar este programa para ver la forma de recibir y transmitir información, (aunque sería mejor que para esto usaras la información que te hemos entregado en los números anteriores).

El programa que estamos entregando es solamente una entretención, pero esperamos que tú le encuentres otras utilidades prácticas a este sistema de transmisión de información.

UNICANDO ADORES

O HARDWARE



DESCRIPCION DEL PROGRAMA

Este programa se basa en el conocido juego Combate Naval, que se juega con dos hojas cuadriculadas. Los papeles, en este caso, serán reemplazados por los computadores. También necesitarás dos televisores, los cuales debes ubicar de tal manera que ningún jugador pueda ver la pantalla de su oponente.

La grilla de la izquierda representa la ubicación de tus barcos y allí además aparecerán los disparos de tu oponente. La de la derecha será donde podrás ver cómo va tu juego sobre la zona de barcos de tu contrincante.

El juego comienza con el primer computador que esté listo para transmitir datos. En este momento verás en la grilla de la derecha un cursor destellante, el cual te indica que te puedes mover hasta donde creas que se encuentra un barco de tu oponente. Una vez allí deberás presionar el botón del joystick para disparar.

Un disparo acertado será indicado con un ruido de explosión y marcado con una X en video inverso.

Un disparo fallido se indicará con una X y un ruido de caída al agua.

Cuando todos los barcos de un jugador estén hundidos, aparecerá un mensaje para el vencedor y uno para el que pierda la partida.

Para recomenzar el juego, basta con presionar el botón del joystick.

DESCRIPCION DEL LISTADO

Líneas 230 a 360:
Iniciación de subrutinas en lenguaje de máquina.
AB espera que los bits recibidos sean distintos de 0 y luego retorna al Basic.

AC espera que los bits recibidos sean 0 y luego retorna al Basic

AD lee el estado de los pines 1 y 2 del joystick B (recibe dos bits) y retorna este valor al Basic.

Líneas 400 a 520:
Muestra el nombre del juego y las grillas en la pantalla

Líneas 560 a 670:
Dibuja los barcos en la grilla de la izquierda:
1 barco de 1 casillero
2 barcos de 2 casilleros
3 barcos de 3 casilleros
2 barcos de 4 casilleros
1 barco de 5 casilleros

Líneas 710 a 730:
Programa los pines 3 y 4 del joystick B como salidas, todos los demás quedan como entradas.

Línea 770:
Comenzar la primera Recepción.

INTERCOMUNICANDO COMPUTADORES

DESARROLLANDO HARDWARE

Línea 810:
Si el primer par de bits recibidos tienen valor 1, el otro computador estaba listo primero, por lo tanto, juega él.

Línea 850:
Si el primer par de bits recibidos tienen valor 3, es turno de jugar.

Líneas 860 y 870:
Si ambos computadores creen que es turno de jugar, transmite un 3 y espera un tiempo al azar. El computador que termina la espera primero,juega primero.

Líneas 910 y 920:
Si el primer valor recibido es 2, es turno de jugar.

Líneas 960 a 990:
Tiempo de espera para el caso de tratar de comenzar ambos al mismo tiempo.

Líneas 1030 y 1040:
Disparo enemigo al agua, mostrar dónde y jugar.

Líneas 1080 a 1180:
Movimiento del joystick, movimiento del cursor en la grilla derecha, sonido del cursor y detección de disparo. El disparo NO es válido si el casillero ha sido usado anteriormente.

Líneas 1220 y 1230:
Transformación de coordenadas para poder transmitir toda la información en un solo byte.

Líneas 1270 a 1320:
Transmisión de las coordenadas del disparo.

Líneas 1360 y 1370:
Si el valor recibido es 3, el disparo cayó al agua. Si se recibe un 2, el disparo acertó en un barco. Si se recibe otro valor,volver a preguntar.

Líneas 1410 a 1450:
Disparo acertado, mostrar disparo, aumentar puntaje. Verificar si todos los barcos del oponente están hundidos, si faltan barcos por hundir, recibir su disparo.

Líneas 1490 a 1560:
El jugador en este computador ganó el juego. Mostrar texto,cambiar colores y hacer sonido de victoria. Si presiona el botón, recomenzar.

Líneas 1600 a 1630:
Disparo al agua, mostrar dónde, y sonido de chapuzón.

Líneas 1670 a 1730:
Recibir coordenadas de disparo enemigo.

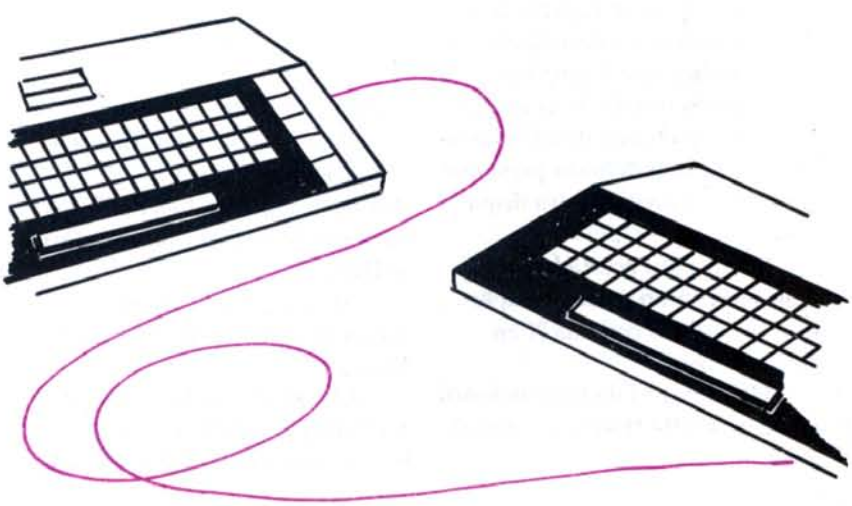
Líneas 1770 y 1780:
Reconstituir coordenadas a partir de ocho bits (1 byte).

Línea 1820:
Si no acertó, jugar.

Líneas 1860 a 1880:
Acertó, mostrar dónde, verificar si quedan más barcos, siendo así, jugar.

Líneas 1920 a 1980:
El jugador en este computador ha perdido la partida. Mostrar texto y sonido de derrota. Si se presiona el botón recomenzar.

Esperamos que con este desarrollo y este juego puedas compartir más tiempo de entretención y aprendizaje con tus amigos Atarianos.



MAPA DE MEMORIA

```
LINEA DEMOSTRACION - CUARENTA CARACTERES
LINEA DEMOSTRACION - CUARENTA CARACTERE
 S
 LINEA DEMOSTRACION - CUARENTA CARACTE
  RES
  LINEA DEMOSTRACION - CUARENTA CARAC
   TERES
   LINEA DEMOSTRACION - CUARENTA CAR
    ACTERES
    LINEA DEMOSTRACION - CUARENTA C
     ARACTERES
     LINEA DEMOSTRACION - CUARENTA
       CARACTERES
      LINEA DEMOSTRACION - CUAREN
       TA CARACTERES
       LINEA DEMOSTRACION - CUAR
        ENTA CARACTERES
        LINEA DEMOSTRACION - CU
         ARENTA CARACTERES
         LINEA DEMOSTRACION -
          CUARENTA CARACTERES
```

por: PEDRO P. CARABALL A.

En esta ocasión completaremos el análisis de las 256 primeras posiciones de memoria, también llamadas: página cero.

Podemos dividir la memoria del computador en 256 páginas de 256 bytes cada una, lo cual nos da el total de la memoria direccionable por un microprocesador de ocho bits.

A continuación de la descripción del número y operación de las posiciones de memoria, encontrarás un programa demostrativo de cómo manejar algunas de estas posiciones.

Es importante refrescar tus conocimientos sobre peek y poke, para que no pierdas nada de la información que te estamos entregando.

POSICION DE MEMORIA: es un lugar en el computador donde se guarda un número, el cual puede tomar valores entre 0 y 255.

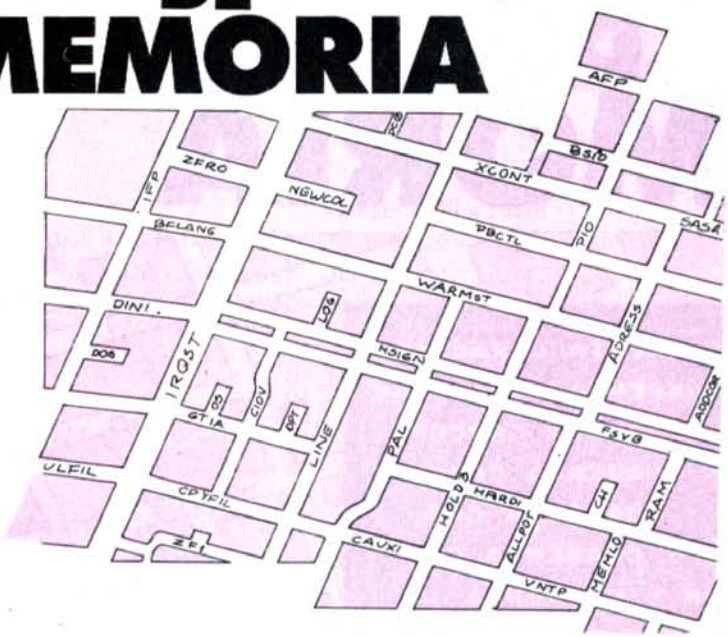
PEEK: examina el número contenido en una determinada posición de memoria.
ej. PRINT PEEK(710)

En este caso,veremos en la pantalla el número que se encuentra en la posición de memoria 710.

POKE: deposita un determinado número en una posición de memoria.
ej. POKE 710,0

En este ejemplo,borramos el número que se encontraba en la posición de memoria 710 y lo reemplazamos por un 0.

MAPA DE MEMORIA



Veamos ahora otra posibilidad:
POKE 712,PEEK(20)

En este caso borramos el número contenido en la posición de memoria 712 y lo reemplazamos con el número que se encontraba en la posición de memoria 20.

Una de las reglas más importantes en el manejo de punteros, es cómo convertirlos correctamente en una variable posible de usar en Basic. Esto se hace mediante la multiplicación del peek mayor por 256 y al resultado se le suma el peek menor.

Por ejemplo, si quisiéramos conocer la posición de la pantalla debemos usar el siguiente método:
? PEEK(89)*256+PEEK(88)

Hay que tener presente que un puntero de este tipo sólo apunta a una parte de la memoria, en la cual se encuentra la información a la que dicho puntero hace referencia.

En el ejemplo anterior, obtuvimos la posición en la cual se encuentra la pantalla, pero si quisiéramos leer la información de la misma tendríamos que basarnos en el siguiente esquema:
?PEEK(PEEK(89)*256+PEEK(88))

Y con este método el resultado impreso en la pantalla será el número que representa al primer caracter que se encuentre en la pantalla (parte superior izquierda). Otra forma de usar r este tipo de puntero sería la siguiente:
POKE(PEEK(89)*256+PEEK(88)),33

Con lo cual aparecerá en el primer byte de la pantalla la representación del número 33 (en este caso una letra A).

128,129 Puntero al comienzo de la Ram disponible. Trabaja conjuntamente con 743 y 744. Es una buena idea modificar este puntero para reservar memoria para datos o rutinas en Assembler.

LAS POSICIONES Y SUS USOS

130,131 Puntero al comienzo de la tabla de nombres de variables del Basic.

132,133 Puntero al fin de la tabla anterior.

134,135 Puntero de la tabla de valores de las variables.

136,137 Puntero al comienzo del programa de Basic codificado.

138,139 Puntero a la línea codificada que está siendo utilizada actualmente por el Basic.

140,141 Dirección de las matrices y variables alfanuméricas. Fin del programa codificado.

144,145 Fin de la memoria utilizada por el Basic.

160,161 Número de línea en la cual se encuentra trabajando el Basic.

186,187 Línea en la cual se detuvo la ejecución del programa Basic.

195 Número de error que causó la detención del programa o la utilización de la instrucción "TRAP".

203,209 No usados por el Basic (disponibles).

212,255 Usados por el paquete de coma flotante.

Les recomendamos estudiar este material y practicar mucho, utilizando los programas adjuntos, ya que veremos muchas más posiciones de memoria en nuestros próximos números.

ATARI EN LINEA

MARCELO WALDBAUM

Para muchos usuarios de computadores, la posibilidad de comunicarse con otras máquinas dentro y fuera del país, se ha convertido en una necesidad.

Esta conección en Computación se llama Telecomunicaciones o Telecomputación y nos permite dialogar a través de Bancos de Datos con usuarios y obtener programas, información económica, financiera y mucho más.

TURBO news se puso como objetivo, promover la utilización de las comunicaciones, así como, también, informar a nuestros lectores las posibilidades con las s que contamos para su mayor aprovechamiento.

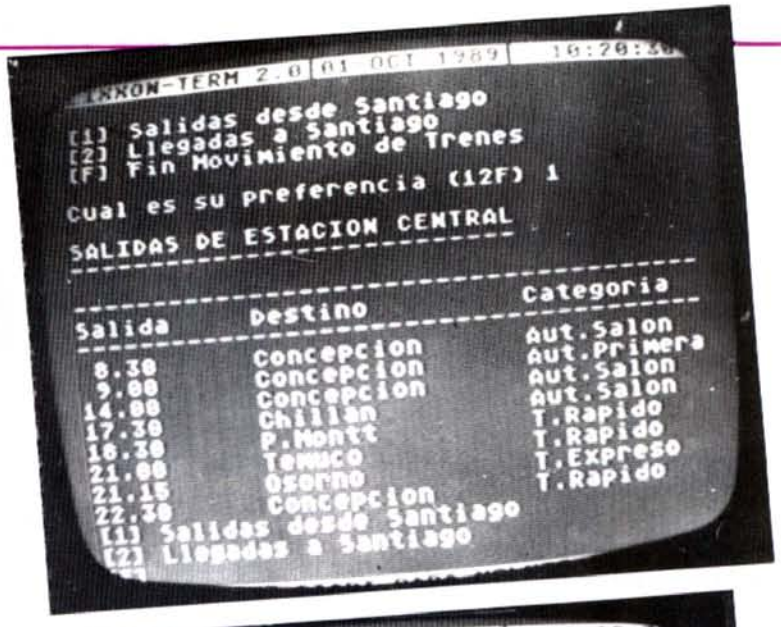
El elemento principal que nos permite trabajar en Telecomunicaciones es un periférico que podemos incorporar a nuestra computadora Atari. Este periférico, conocido con el nombre de Modem, convierte datos de nuestro computador, en señales transmisibles por una línea telefónica y asimismo recibe señales telefónicas y las convierte en información que el computador puede interpretar.

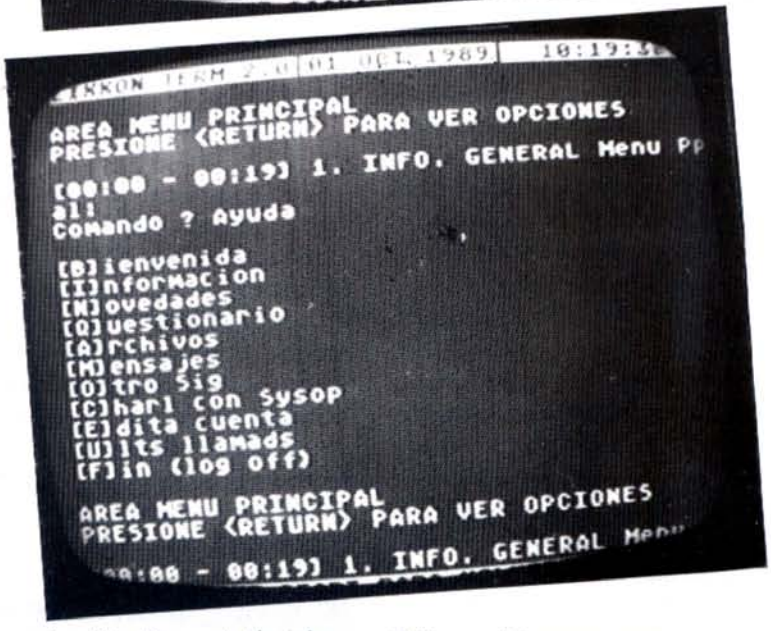
Para establecer una comunicación con otro usuario, es necesario que ambos posean un Modem de similares características y una línea telefónica.

En comunicaciones podemos trabajar con dos normas o sistemas. La norma CCITT y la BELL. Esto significa que si queremos establecer una comunicación con otro computador, ambos Modems deben trabajar en la misma norma. Los Modems que podemos obtener en el mercado pueden ser de una sola norma o bien, binorma. Si son, por ejemplo, sólo BELL, con ellos nos podremos comunicar con otros usuarios o bancos de datos que funcionen en BELL, pero si son BINORMA, podemos acceder a todos los bancos y usuarios que trabajen con Modems.

Otro aspecto importante para tomar en cuenta, es la velocidad de transferencia con la cual podemos transmitir. En Telecomunicaciones, la velocidad de

ATARI EN LINEA EN LINEA EN LINEA EN LINEA





transferencia se mide en Baudios o bits por segundo. Existen Modems que trabajan a 300, 1200, y 2400 baudios para nuestro computador ATARI, pero la velocidad standard son los 300 Baudios, pues es la más utilizada en los Bancos de Datos a los cuales podemos acceder.

También es interesante tomar en cuenta a aquellos Modems que tienen la posibilidad de discar automáticamente, como así también,recibir comunicaciones sin la necesidad de que el usuario esté presente en la conección. Cuando un Modem tiene la posibilidad de discar automáticamente, decimos que es un Modem Auto-Dial. Con él,al querer establecer una comunicación, lo único que tenemos que hacer es ingresar por el teclado el número telefónico del otro usuario y es el mismo Modem el que hace todo el trabajo hasta lograr establer el diálogo.

Cuando un Modem es capaz de recibir automáticamente llamadas, decimos que es un Modem Auto-answer. Estos Modems al estar conectados con las líneas telefónicas, tienen la posibilidad de reconocer cuando el teléfono está llamando, ante lo cual le informa al computador que alguien está queriendo establecer un llamado y es el computador quien habilita la comunicación. Con este tipo de Modems se desarrollan los Bancos de Datos. Cuando un Banco de Datos esta en funcionamiento, generalmente el operador del mismo no se encuentra frente al computador del Banco. Es este, a traves de su software, el que está recibiendo llamados y el que permite o no ingresar al Banco según el usuario posea o no una clave de acceso que lo habilite para utilizar dicho Banco de Datos.

Como ejemplo de una comunicación hemos ingresado al Banco de Datos del Atari Club de Chile y publicamos en este artículo la comunicación que hemos establecido. En las siguientes ediciones de tu revista TURBO news, seguiremos investigando el tema de las comunicaciones, ingresando a los demás Bancos de Datos existentes en el mercado.

MichTron's Multi-User BBS.
Version 2.10
Copyright (c) 1985,1986,1987
MichTron Inc.
Program By Timothy Puryes

BIENVENIDOS AL BBS ATARI COELSA

DESEA VER:
[A] LA PRESENTACION
[B] NOVEDADES (21.08.89)
[N] Ninguna
Cuál es su preferencia (ABN) A

Bienvenidos al...

BBS Atari Coelsa
ASCII -300-1200bps - No Parity

```
** ** **
** ** **
** ** **
** ** **
** ** **
** ** **
*** ** ***
***  **  ***
***  **  ***
```

B B S!
BBS especial para socios
del Club ATARI COELSA.
WizOp: Sr.Esteban Alarcón.
SysOp: Sr.Jacob Pichara I.
<<Horario de Atención>>
De Martes a Domingo las 24:00 hrs.
Lunes desde las 15:00 hrs.!
Logged on Modem 1

Nombre: **MPM**
Clave : #######

Bienvenido al BBS ATARI COELSA.
MPM, Registrado
1 Llamada(s), la última vez Miérc., Jul 25, 1989 en 12:59
Hoy le quedan 20 minutos en su cuenta.
1024 mensajes
1283 downloads
AREA MENU PRINCIPAL
PRESIONE <RETURN> PARA VER OPCIONES

[00:00 - 00:19] 1. INFO. GENERAL Menu Ppal:
Comando ? Ayuda
[B]ienvenida
[I]nformación
[N]ovedades
[Q]uestionario
[A]rchivos
[M]ensajes
[O]tro Sig
[C]harl con Sysop
[E]dita cuenta
[U]lts llamads
[F]in (log off)

AREA MENU PRINCIPAL
PRESIONE <RETURN> PARA VER OPCIONES
00:00 - 00:19] 1. INFO. GENERAL Menu Ppal:
Comando ? Información
Existe información en los siguentes temas:
[1] Indicadores Económicos.
[2] Ofertas Centro Atari.
[3] Teléfonos de interés.
[4] Movimiento de trenes
[5] Otros BBS.
[6] Teatro Municipal
[F] Fin de la información.
Cuál es su preferencia (123456F) 1
Existe información para los siguientes Indicadores Económicos.
[1] Valor de la U.F. del mes.
[2] Dólar Oficial del mes.
[3] I.P.C. últimos 12 meses.
[4] Interés Bancario.
[5] Indice Valor Promedio
[6] Unidad Tributaria Mensual
[7] Costos Previsionales de AFP
[F] Fin Infomación Indicadores Económicos.
Cuál es su preferencia (1234567F) 1
Los valores de la U.F. para el período: 10 de Agosto al 15 de Agosto de 1989.
10/08/89 $4.971.98
11/08/89 $4.974.84
12/08/89 $4.977.71
13/08/89 $4.980.57
14/08/89 $4.983.44
15/08/89 $4.986.31
Cuál es su preferencia (123456F) 4
[1] Salidas desde Santiago
[2] Llegadas a Santiago
[F] Fin Movimiento de Trenes
Cuál es su preferencia (12F) 1
SALIDAS DE ESTACION CENTRAL

Salida	Destino	Categoría
8.30	Concepción	Aut.Salón
9.00	Concepción	Aut.Primera
14.00	Concepción	Aut.Salón
17.30	Chillan	Aut.Salón
18.30	P.Montt	T.Rápido
21.00	Temuco	T.Rápido
21.15	Osorno	T.Expreso
22.30	Concepción	T.Rápido

[1] Salidas desde Santiago
[2] Llegadas a Santiago
[F] Fin Movimiento de Trenes
Cuál es su preferencia (12F) F
Fin Movimiento de trenes

Existe información en los siguentes
temas:
[1] Indicadores Económicos.
[2] Ofertas Centro Atari.
[3] Teléfonos de interés.
[4] Movimiento de trenes
[5] Otros BBS.
[6] Teatro Municipal
[F] Fin de la información.
Cuál es su preferencia (123456F) 5
A continuación les entrego una lista de otros BBS
que están en funcionamiento en Chile.

BBS EN CHILE

BBS Atari Coelsa 5569132
BBS Hacker's 2816136
En la semana desde las 20.00
Los Sábados y domingos todo el
día.
BBS Wizard 2244632
Lunes a Jueves
20.45 - 00.00 Hrs.
Viernes-Sabados-Domingos
20.45 - 02.00 Hrs.
Cuál es su preferencia (123456F) F
Fin de la información
AREA MENU PRINCIPAL
PRESIONE <RETURN> PARA VER OPCIONES
[00:04 - 00:15] 1. INFO. GENERAL Menu Ppal:
Comando ? Ayuda
[B]ienvenida
[I]nformación
[N]ovedades
[Q]uestionario
[A]rchivos
[M]ensajes
[O]tro Sig
[C]harl con Sysop
[E]dita cuenta
[U]lts llamads
[F]in (log off)
AREA MENU PRINCIPAL
PRESIONE <RETURN> PARA VER OPCIONES
[00:04 - 00:15] 1. INFO. GENERAL Menu Ppal:
Comando ? Mensajes
AREA DE MENSAJES -
PRESIONE <RETURN> PARA VER COMANDOS
[00:05 - 00:14] 1. INFO. GENERAL Menú mens:
Comando ? Ayuda
[R]evs mens
[L]eer mens
[V]er individual
[D]ejar mens
[S]olo mens.nvo
[M]enu Ppal
[F]in (log off)
[00:05 - 00:14] 1. INFO. GENERAL Menú mens:
Comando ? Leer mensajes
Mensajes abarcan de 2522 a 3547.
En qué núm comienza ? 2522
En que Sig(s) ver mens (? para listar) [1-15] : ?
[*] =Sigs permitidos [D] = Su Default
1 [D] INFO. GENERAL
2 [*] ATARI 8 BITS
3 [ ] ATARI 16 BITS
4 [*] NOTICIAS/VARIOS
5 [*] COMPUTACION
6 [ ] CENTROS ATARI
7 [ ] COLEGIOS
8 [ ] No asignado
9 [*] MUSICA
10 [*] FINANCIERO
11 [ ] No asignado
12 [ ] No asignado
13 [ ] No asignado
14 [ ] No asignado
15 [ ] CONFIDENCIAL
En qué Sig(s) ver mens (? para listar) [1-15] : 2
Mostrando Mensaj.Sig(s)2
Pausa entre mensajes (Y,N,Q) [Y] ? No
Pausa si hay carta para/de Ud (Y,N,Q) [Y] ? Si
Sólo muestra mens. dirigidos a Ud (Y,N,Q) [N] ? No
Mensaje : 2554 [Abre] 05/16/89 21:20
De : Fabiola Concha
Para : Dennis Cangme
Materia : RBK66613
Sig(s) : 2 (ATARI 8 BITS)
Hay 1 respta

Dennis:
Llegó el paquete de EEUU con:
-IHUMANOYD-POLICLORATRON-MEGATRANS-
-EXENTER-LPOSTAL EXO-
y un montón más, espero mañana tener el sec 348
de telec.
atte.
SATAN.inc.

Mensaje : 2563 [Abre] 05/16/89 22:26
De : Dennis Kangme
Para : Fabiola Concha (x)
Materia : #2554 RBK66613
Sig(s) : 2 (ATARI 8 BITS)

ok.viejo espero el sec 348 y espero una
laaaaaaaaaarga charla contigo de algo muy SERIO.

Dennis

Nota: Los textos que se encuentran destacados, corresponden a nuestra respuestas.

ASSEMBLER

En esta sección del curso de Assembler, queremos completar la explicación de algunas de las instrucciones que nos faltan analizar, con el objetivo de poder ingresar en el terreno de la programación de rutinas completas en lenguaje de máquina.

MARCELO WALDBAUM

En las ediciones anteriores de tu Revista TURBO news, te explicamos las aplicaciones de los Registros A, X e Y. Sobre estos dos últimos registros, podemos realizar las siguientes instrucciones Assembler:

INSTRUCCION INY:

Al encontrar el microprocesador la instrucción INY, al contenido del Registro Y se le suma 1 y el resultado se vuelve a almacenar en este registro. Esta instrucción es muy útil, sobre todo cuando queremos realizar ciclos en los cuales debemos contar la cantidad de veces que estos tienen que realizarse.

Registro Y 00000010 estado inicial del registro.

Instrucción INY.

Registro Y 00000011 estado final del registro.

INSTRUCCION DEY:

Esta instrucción toma el valor del registro Y, le resta 1 y el resultado es almacenado nuevamente en Y. Su operatoria es similar a la instrucción INY:

Registro Y 00000011 estado inicial del registro.

Instrucción DEY.

Registro Y 00000010 estado final del registro.

Para el registro X también existen las instrucciones de incremento y decremento en 1. En caso de querer incrementar el registro X,debemos aplicar la instrucción INX.

INSTRUCCION INX:

Al encontrar el procesador la instrucción INX, al contenido del registro X se le suma 1.

Registro X 01010110 estado inicial del registro.

Instrucción INX.

Registro X 01010111 estado final del registro.

LECCION 4

INSTRUCCION DEX:

Con esta instrucción, el 6502 decrementa en 1 al registro X.

Registro X 10010011 estado inicial del registro.

Instrucción DEX.

Registro X 10010010 estado final del registro.

Es importante mencionar que estas instrucciones no afectan al contenido del acumulador. Pero para éste no existen instrucciones que lo incrementen o decrementen en uno. Para realizar un incremento del registro A podemos utilizar la instrucción ADC,ya analizada en las ediciones anteriores de TURBO news, o bien ejecutar un traspaso del registro A al registro X, e incrementar el X, volviendo a transpasar el registro X al acumulador. Si desarrollamos el incremento del registro A con la instrucción ADC tendremos que ingresar las siguientes líneas de programa:

```
100  .OPT OBJ
110  * = $4000
120  CLC
130  LDA #$41
140  ADC #$01
150  BRK
```

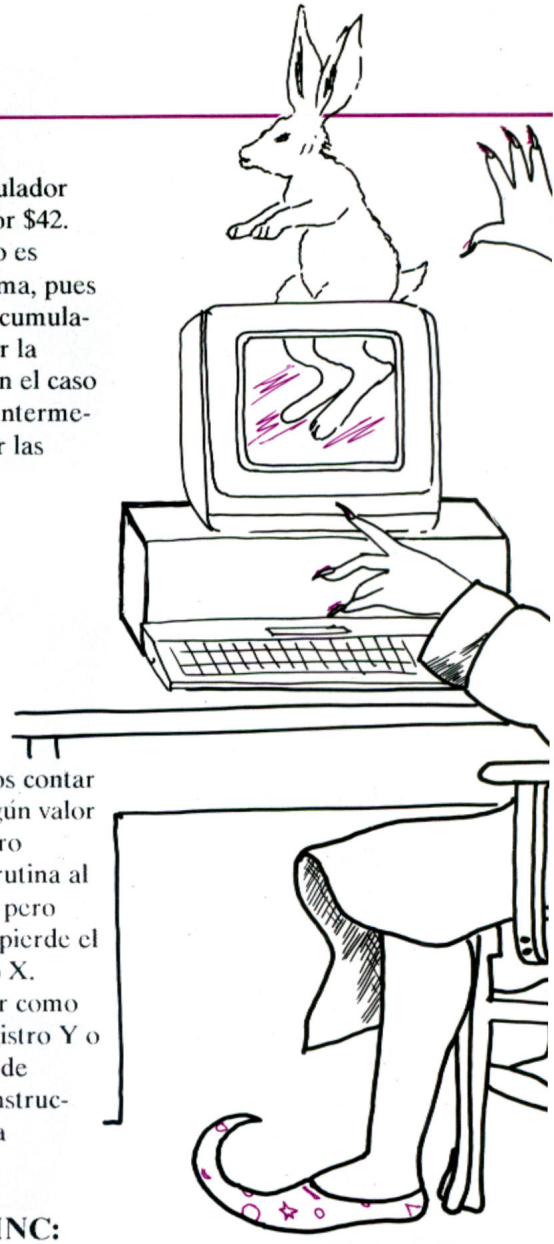
En este caso el acumulador queda definido con el valor $42. Claro que este ejemplo no es aplicable en algún programa, pues podemos para definir al acumulador con el byte $42 utilizar la instrucción LDA #$42. En el caso de utilizar como registro intermedio al X,debemos ejecutar las instrucciones siguientes:

```
100  .OPT OBJ
110  * = $4000
120  LDA #$41
130  TAX
140  INX
150  TXA
160  BRK
```

En este caso, debemos contar con un registro X sin ningún valor que nos interese en nuestro programa, pues con esta rutina al acumulador se le suma 1, pero como efecto colateral, se pierde el valor anterior del registro X. También podemos utilizar como registro intermedio al registro Y o bien, cualquier dirección de memoria, utilizando las instrucciones que explicaremos a continuación:

INSTRUCCION INC:

La instrucción INC es similar a las instrucciones INX e INY, pero a diferencia de éstas, trabaja directamente con posiciones de nuestro Mapa de Memoria. Es decir, que si el procesador Atari se encuentra con la instrucción INC $401A, el resultado de esta operación, es el incremento en uno de la dirección de memoria $401A. Al igual que las instrucciones INX e INY, ésta no modifica el contenido del acumulador una vez ejecutada.



CON EL ASSEMBLER LE PUEDE SACAR HASTA CONEJOS AL COMPUTADOR

Direccion $401A |00100000| estado inicial.

Instrucción INC $401A.

Dirección $401A |00100001| estado final.

Si queremos incrementar el acumulador, utilizando como registro intermedio una dirección de memoria debemos ingresar las siguientes líneas de programa:

```
100   .OPT OBJ
110   *=$4000
120   LDA #$41
130   STA $5000
140   INC $5000
150   LDA $5000
160   BRK
```

Una vez analizadas las instrucciones de incremento de registros, estudiaremos ahora cómo tomar decisiones en lenguaje Assembler.

En Basic, el manejo de la toma de decisiones es sencillo. Para trabajar con ellas, contamos con la instrucción IF ... THEN. Si queremos sumarle 10 a la variable PUNTAJE, siempre que el contenido de la variable DISPARO sea 0, lo único que debemos hacer es ingresar la línea BASIC:

```
100 IF DISPARO=0 THEN
PUNTAJE=PUNTAJE+10
```

En Assembler, este trabajo es un poco más complicado. Para la toma de decisiones es necesario conocer el funcionamiento del indicador de resultado cero, que se encuentra contenido en el registro de estado del procesador. En la edición anterior de este curso, hemos explicado el funcionamiento del indicador de acarreo. Cuando ejecutamos una suma y el resultado excede los 8 bits, éste se pone en 1. Si esto no ocurre, toma el valor 0. El indicador de resultado cero funciona en forma similar a este proceso. Cuando el resultado de una operación que el procesador ejecuta es cero, el indicador de resultado cero toma el valor 1, caso contrario se define en 0.

Utilizando este indicador de resultado cero, existen dos instrucciones de bifurcación o toma de decisión: BEQ y BNE.

INSTRUCCION BEQ:

La instrucción BEQ, consulta el contenido del indicador de resultado cero. Si éste está en 1, es decir, la última operación que realizó el procesador produjo un resultado cero, salta a la dirección que esta instrucción tiene contenida en su operando. Por ejemplo:

```
100   .OPT OBJ
110   *=$4000
120   LDA #$00
130   BEQ IMPRIMOB
140 IMPRIMOA
150   LDA #$41
160   JSR $F2B0
170   JMP FIN
180 IMPRIMOB
190   LDA #$42
200   JSR $F2B0
210 FIN
220   RTS
```

Cuando el procesador Atari encuentra esta rutina, con la línea 120 define al acumulador en cero y automáticamente el indicador de resultado cero se define en 1. Luego con la instrucción BEQ, salta al rótulo IMPRIMOB por encontrarse en 1 el flag de resultado cero. Es aquí, donde se imprime en la pantalla el caracter B y concluye esta rutina con un RTS. Es decir, que las líneas 150, 160 y 170 nunca son accedidas por el procesador.

INSTRUCCION BNE:

Esta instrucción de bifurcación es similar a la anterior, pero salta en caso de que el indicador de resultado cero, se encuentre en el valor 0. Por lo tanto, si la última instrucción no arrojó un resultado nulo, la instrucción BNE tiene efecto en la bifurcación.

Si en el programa anterior hubiésemos cargado al acumulador con el valor #$23 en la instrucción 120 y en la línea 130 un BNE IMPRIMOB, obtendríamos el mismo resultado en cuanto a la impresión de la letra B en la pantalla.

Hemos visto cómo realizar bifurcaciones dependiendo de si la ultima operación arrojó un valor cero o distinto de cero, pero en la práctica, muchas veces necesitamos realizar saltos condicionales dependiendo de si determinada variable es igual o no a determinado valor. En BASIC nuevamente contamos con la instrucción IF ... THEN y en Assembler con las instrucciones de comparación de registros.

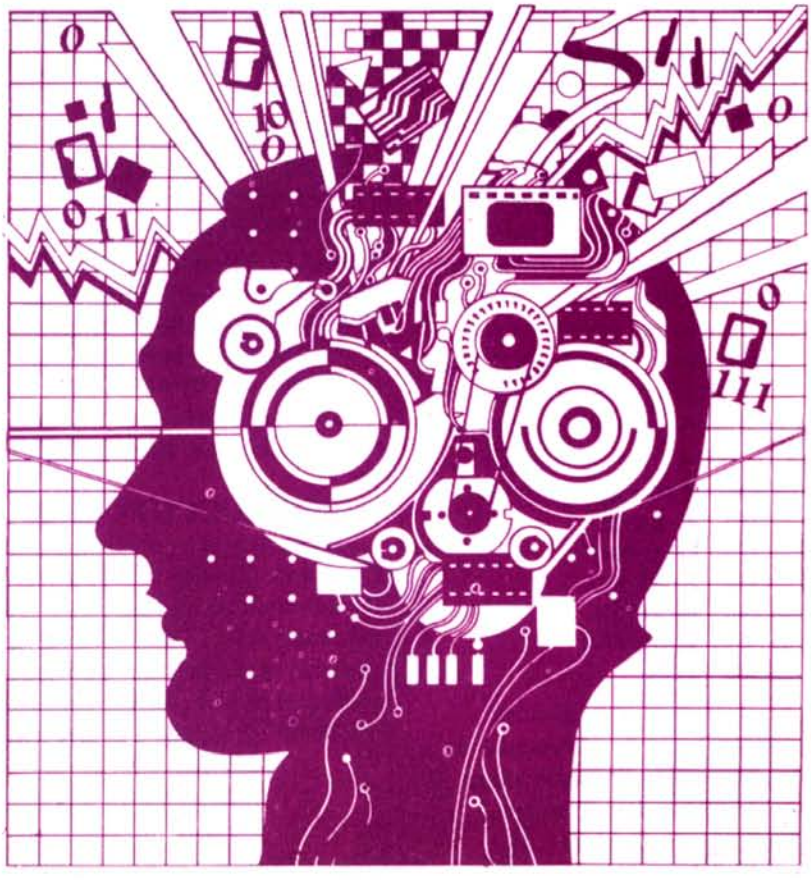
INSTRUCCION CMP:

La instrucción CMP compara dos valores: uno de ellos es el contenido en el acumulador y el otro puede ser una posición de la memoria de nuestro computador. Al realizar la comparación no se modifican los registros, pero si los indicadores que se encuentran en el registro de estado del 6502. En realidad, el procesador para comparar dos valores resta uno del otro. Si son iguales, al restarlos el resultado nos da 0 y por lo tanto el indicador de resultado cero se define en 1. Si son distintos el flag de cero se coloca en 0.

Utilizando este sistema de comparación, una vez ejecutada la instrucción CMP, podemos realizar un salto condicional BEQ o bien BNE.

También es posible comparar los registros X e Y con posiciones de la memoria. En el caso del registro X, la instrucción a utilizar es CPX y con el registro Y CPY. Su funcionamiento es similar a la instrucción CMP pero en vez de actuar con el acumulador, trabajan con X e Y.

En este artículo hemos completado algunos conceptos necesarios para el desarrollo de programas complejos en Assembler. Incorporamos también en la página siguiente, una tabla de referencia de las instrucciones que nuestro procesador puede interpretar. Esta tabla es muy útil, cuando queremos saber si determinada instrucción afecta o no a los indicadores que se encuentran contenidos en el registro de estado del microprocesador y qué bytes hexadecimales representan cada una de las instrucciones.



INSTRUCCIONES DEL PROCESADOR 6502

Mnemonic Code	Operation Performed	Status Flags N	V		B	D	I	Z	C
ADC	Add memory to accumulator with carry	X	X					X	X
AND	AND memory with accumulator	X						X	
ASL	Shift left one bit (memory or accum.)	X						X	X
BCC	Branch on carry clear (If C=0)								
BCS	Branch on carry set (If C=1)								
BEQ	Branch on result zero (If Z=1)								
BIT	Test bits in accumulator with memory	X	X					X	
BMI	Branch on result minus (If N=1)								
BNE	Branch on result not zero (If Z=0)								
BPL	Branch on result plus (If N=0)								
BRK	Force Break						1		
BVC	Branch on overflow clear (If V=0)								
BVS	Branch on overflow set (If V=1)								
CLC	Clear carry flag								0
CLD	Clear decimal mode					0			
CLI	Clear interrupt disable flag						0		
CLV	Clear overflow flag		0						
CMP	Compare memory and accumulator	X						X	X
CPX	Compare memory and index X	X						X	X
CPY	Compare memory and index Y	X						X	X
DEC	Decrement memory by one							X	X
DEX	Decrement index X by one							X	X
DEY	Decrement index Y by one							X	X
EOR	Exclusive OR memory with accumulator							X	X
INC	Increment memory by one							X	X
INX	Increment index X by one							X	X
INY	Increment index Y by one							X	X
JMP	Jump to new location								
JSR	Jump to new location save rtn. add.								
LDA	Load accumulator from memory							X	X
LDX	Load index X from memory							X	X
LDY	Load index Y from memory							X	X
LSR	Shift right one bit (memory or accum.)	0						X	X
NOP	No operation								
ORA	OR memory with accumulator	X						X	
PHA	Push accumulator on stack								
PHP	Push processor status on stack								
PLA	Pull accumulator from stack							X	X
PLP	Pull processor status from stack	X	X	X	X	X	X	X	X
ROL	Rotate one bit left (mem. or accum.)	X						X	X
ROR	Rotate one bit right (mem. or accum.)	X						X	X
RTI	Return from interrupt	X	X	X	X	X	X	X	X
RTS	Return from subroutine								
SBC	Subtract memory and borrow from accum.	X	X					X	X
SEC	Set carry flag								1
SED	Set decimal mode					1			
SEI	Set interrupt disable flag						1		
STA	Store accumulator in memory								
STX	Store index X in memory								
STY	Store index Y in memory								
TAX	Transfer accumulator to index X	X						X	
TAY	Transfer accumulator to index Y	X						X	
TSX	Transfer stack pointer to index X	X						X	
TXA	Transfer index X to accumulator	X						X	
TXS	Transfer index X to stack pointer								
TYA	Transfer index Y to accumulator	X						X	

Mnemonic Code	Op Codes Accumulator	Immediate	Zero Page	Zero Page, X	Zero Page, Y	Absolute	Absolute, X	Absolute, Y	Implied	Relative	Indexed Indirect	Indirect Indexed	Indirect
ADC	–	69	65	75	–	6D	7D	79	–	–	61	71	–
AND	–	29	25	35	–	2D	3D	39	–	–	21	31	–
ASL	0A	–	06	16	–	0E	1E	–	–	–	–	–	–
BCC	–	–	–	–	–	–	–	–	–	90	–	–	–
BCS	–	–	–	–	–	–	–	–	–	B0	–	–	–
BEQ	–	–	–	–	–	–	–	–	–	F0	–	–	–
BIT	–	–	24	–	–	2C	–	–	–	–	–	–	–
BMI	–	–	–	–	–	–	–	–	–	30	–	–	–
BNE	–	–	–	–	–	–	–	–	–	D0	–	–	–
BPL	–	–	–	–	–	–	–	–	–	10	–	–	–
BRK	–	–	–	–	–	–	–	–	00	–	–	–	–
BVC	–	–	–	–	–	–	–	–	–	50	–	–	–
BVS	–	–	–	–	–	–	–	–	–	70	–	–	–
CLC	–	–	–	–	–	–	–	–	18	–	–	–	–
CLD	–	–	–	–	–	–	–	–	D8	–	–	–	–
CLI	–	–	–	–	–	–	–	–	58	–	–	–	–
CLV	–	–	–	–	–	–	–	–	B8	–	–	–	–
CMP	–	C9	C5	D5	–	CD	DD	D9	–	–	C1	D1	–
CPX	–	E0	E4	–	–	EC	–	–	–	–	–	–	–
CPY	–	C0	C4	–	–	CC	–	–	–	–	–	–	–
DEC	–	–	C6	D6	–	CE	DE	–	–	–	–	–	–
DEX	–	–	–	–	–	–	–	–	CA	–	–	–	–
DEY	–	–	–	–	–	–	–	–	88	–	–	–	–
EOR	–	49	45	55	–	4D	5D	59	–	–	41	51	–
INC	–	–	E6	F6	–	EE	FE	–	–	–	–	–	–
INX	–	–	–	–	–	–	–	–	E8	–	–	–	–
INY	–	–	–	–	–	–	–	–	C8	–	–	–	–
JMP	–	–	–	–	–	4C	–	–	–	–	–	–	6C
JSR	–	–	–	–	–	20	–	–	–	–	–	–	–
LDA	–	A9	A5	B5	–	AD	BD	B9	–	–	A1	B1	–
LDX	–	A2	A6	–	B6	AE	–	BE	–	–	–	–	–
LDY	–	A0	A4	B4	–	AC	BC	–	–	–	–	–	–
LSR	4A	–	46	56	–	4E	5E	–	–	–	–	–	–
NOP	–	–	–	–	–	–	–	–	EA	–	–	–	–
ORA	–	09	05	15	–	0D	1D	19	–	–	01	11	–
PHA	–	–	–	–	–	–	–	–	48	–	–	–	–
PHP	–	–	–	–	–	–	–	–	08	–	–	–	–
PLA	–	–	–	–	–	–	–	–	68	–	–	–	–
PLP	–	–	–	–	–	–	–	–	28	–	–	–	–
ROL	2A	–	26	36	–	2E	3E	–	–	–	–	–	–
ROR	6A	–	66	76	–	6E	7E	–	–	–	–	–	–
RTI	–	–	–	–	–	–	–	–	40	–	–	–	–
RTS	–	–	–	–	–	–	–	–	60	–	–	–	–
SBC	–	E9	E5	F5	–	ED	FD	F9	–	–	E1	F1	–
SEC	–	–	–	–	–	–	–	–	38	–	–	–	–
SED	–	–	–	–	–	–	–	–	F8	–	–	–	–
SEI	–	–	–	–	–	–	–	–	78	–	–	–	–
STA	–	–	85	95	–	8D	9D	99	–	–	81	91	–
STX	–	–	86	–	96	8E	–	–	–	–	–	–	–
STY	–	–	84	94	–	8C	–	–	–	–	–	–	–
TAX	–	–	–	–	–	–	–	–	AA	–	–	–	–
TAY	–	–	–	–	–	–	–	–	A8	–	–	–	–
TSX	–	–	–	–	–	–	–	–	BA	–	–	–	–
TXA	–	–	–	–	–	–	–	–	8A	–	–	–	–
TXS	–	–	–	–	–	–	–	–	9A	–	–	–	–
TYA	–	–	–	–	–	–	–	–	98	–	–	–	–

RANKING DEL MES

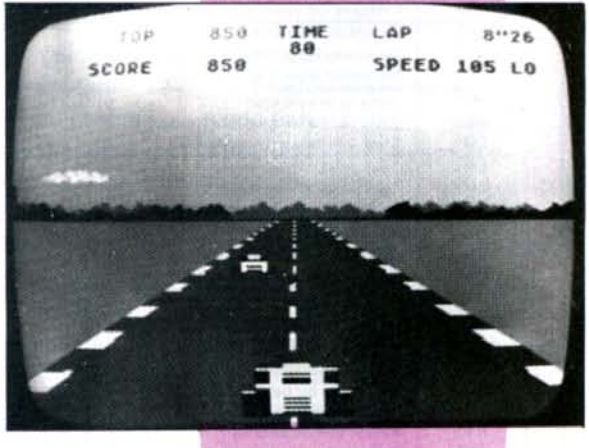
POS. DEL MES	POS. MES ANTERIOR	TITULO
1	2	MONTEZUMA'S REVENGE
2	10	INTERNATIONAL KARATE
3	5	NINJA
4	1	SCREEMINGS WINGS 1942
5	8	POLE POSITION
6	-	BMX SIMULATOR
7	23	FUTBOL II
8	4	RIVER RAID
9	7	BRUCE LEE
10	-	MONTEZUMA
11	-	BLUE MAX
12	15	LEADER BOARD GOLF
13	3	GREAT AMERICAN RACE
14	-	NINJA MASTER
15	22	STAR RAIDERS II
16	20	OIL'S WELL
17	-	RESCUE ON FRACTALUS
18	13	SWAT
19	14	GHOSTBUSTERS
20	12	ELEKTRAGLIDE
21	6	FLAK
22	18	BOULDER DASH II
23	-	MARIO BROSS.
24	-	FIGHTER PILOT II
25	17	PACMAN

ASCENSO CONSTANTE DESCENSO

Continuando con la entrega de rankings mensuales, entregamos en este número el correspondiente al mes de Agosto, obtenido en base a las estadísticas de ventas de cassettes Turbo Software en todo Chile. Recuerda que tus preferencias también serán tenidas en cuenta, para lo cual puedes escribir a Holanda 2456, Providencia, con los juegos de tu elección.

POLE POSITION

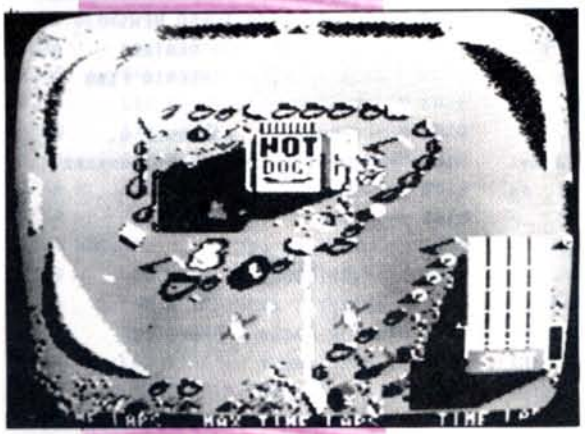
En esta famosa competición de Fórmula 1, se ponen a prueba tus reflejos y habilidad para conducir a velocidades de hasta 195 Millas por hora. Antes de ser habilitado para largar, deberás pasar con éxito una prueba de clasificación. Dispones de 90 segundos, pero debes completar el recorrido en menos de 74 para participar en la carrera. Superada la prueba, viene la emoción principal. Durante la carrera, no sólo deberás tratar de obtener el mejor tiempo, sino también, sobrepasar a los demás competidores. Tienes un tiempo máximo para cumplir cada vuelta. De no conseguirlo, serás automáticamente descalificado. Cuentas con dos velocidades: lenta, moviendo el joystick hacia adelante y rápida moviéndolo hacia atras. El botón del mismo funciona como freno. Un desafío para experimentados del volante.



SONIDO	6.4
GRAFICACION	6.6
ADICCION	6.6
PRESENTACION	6.5
PROMEDIO	6.525

BMX SIMULATOR

SONIDO	6.7
GRAFICACION	6.8
ADICCION	6.7
PRESENTACION	6.6
PROMEDIO	6.7

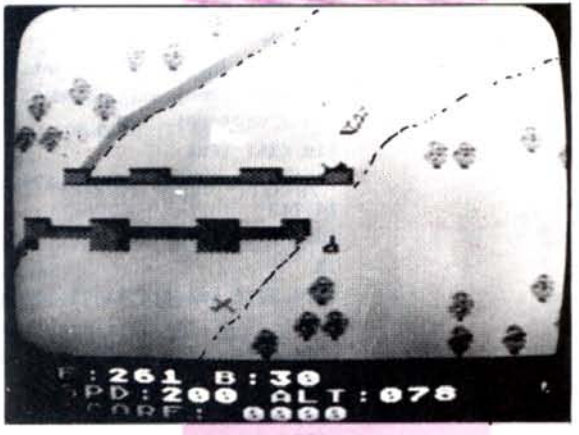


El objetivo de este juego es conducir una bicicleta por sinuosos circuitos, plagados de rampas, curvas cerradas y demás obstáculos, participando en una carrera en la que deberás enfrentar, ya sea al computador o bien a un amigo. Cada pista (son siete en total), posee un tiempo mínimo para ser recorrida, sin el cual no pasarás a la pista siguiente. Las mismas son de dificultad creciente y debes esforzarte al máximo para mantenerte en competencia. Podrás además ver la repetición de una carrera al ser concluida y los accidentes en camara lenta. Su uso es muy sencillo moviendo el joystick para doblar y manteniendo el botón del mismo oprimido para pedalear.

BLUE MAX

El BLUE MAX es un juego de combate aéreo tridimensional. Tu misión consiste en destruir la mayor cantidad de aviones enemigos durante los combates aéreos, incluyendo los blancos en tierra. Para ello, deberás efectuar violentos bombardeos o bien ametrallarlos con vuelos rasantes. Los blancos consisten en puentes, edificios, aviones, tanques, baterias anti-aéreas, vehículos y barcos. Podrás aterrizar en los aeropuertos aliados para reabastecimiento de combustible y reparación de posibles daños (fugas de combustible, ametralladoras dañadas, dispositivos de bombardeo defectuosos, etc). Cuentas con un indicador al pie de la pantalla que te avisará cuándo un avión enemigo se encuentra a tu misma altura y puedes, por lo tanto, ser alcanzado por sus balas.

MAURO PIERESSA S.



SONIDO	6.5
GRAFICACION	6.6
ADICCION	6.8
PRESENTACION	6.7
PROMEDIO	6.65

Atari por dentro

```
100 REM ****************************
110 REM *    REVISTA TURBO NEWS    *
120 REM *     ATARI POR DENTRO     *
130 REM *   EJEMPLO DE MOVIMIENTO  *
140 REM *   FINO CON INTERRUPCION  *
150 REM *   DE BLANQUEO VERTICAL   *
160 REM *                          *
170 REM *    PEDRO P. CARABALL A.  *
180 REM ****************************
190 REM
200 RESTORE :C=0
210 FOR I=1536 TO 1673
220 READ B:C=C+B
230 POKE I,B
240 NEXT I
250 IF C<>14965 THEN ? "ERROR DE DATA"
:END
260 FOR I=1694 TO 1769:POKE I,0:NEXT I
270 POKE 710,0:DIM A$(75)
280 ? "INGRESA EL TEXTO A DESPLEGAR"
290 ? "  (MAXIMO 75 CARACTERES)"
300 ? ">";:INPUT #16;A$
310 P=PEEK(88)+PEEK(89)*256
320 FOR I=1 TO LEN(A$)
330 POSITION 0,0:? CHR$(27);A$(I,I)
340 POKE 1693+I,PEEK(P):? PEEK(P)
350 NEXT I
360 POKE P,0
370 JMP=ADR("PPCA")
380 POKE JMP,104:POKE JMP+1,76
390 POKE JMP+2,0:POKE JMP+3,6
400 X=USR(JMP)
410 LIST :END
1000 DATA 173,14,212,72,120,169,0,141,
14,212
1010 DATA 169,6,141,35,2,169,26,141,34
,2
1020 DATA 104,141,14,212,88,96,173,49,
2,201
1030 DATA 6,240,45,173,48,2,133,203,24
,105
1040 DATA 3,141,136,6,173,49,2,133,204
,105
1050 DATA 0,141,137,6,160,0,177,203,20
0,201
1060 DATA 65,208,249,169,130,145,203,1
41,48,2
1070 DATA 200,169,6,145,203,141,49,2,2
06,129
1080 DATA 6,16,37,169,7,141,129,6,238,
133
1090 DATA 6,208,3,238,134,6,173,133,6,
201
1100 DATA 234,208,17,173,134,6,201,6,2
08,10
1110 DATA 169,138,141,133,6,169,6,141,
134,6
1120 DATA 173,129,6,141,4,212,76,95,22
8,8
1130 DATA 112,112,86,138,6,1,255,255,0
```

Listado 1

```
0100 ****************************
0110 *     REVISTA TURBO NEWS     *
0120 *      ATARI POR DENTRO      *
0130 * ANTIC Y EL MOVIMIENTO FINO *
0140 *                            *
0150 *    PEDRO P. CARABALL A.    *
0160 ****************************
0170 ;
0180 ;
0190      .OPT NO LIST
0200      .OPT OBJ
0210 ;
0220 ORG =   $0600    ; ORIGEN
0230      *=  ORG
0240 ;
0250      LDA $D40E   ; INTERRUPCIONES
0260      PHA         ; A LA PILA
0270      SEI         ; NO INTERRUMPA
0280      LDA #$00    ; V.B.I.-D.L.I.
0290      STA $D40E   ; NADIE!
0300      LDA # >VBI  ; CAMBIAR
0310      STA $0223   ; VECTOR
0320      LDA # <VBI  ; INMEDIATO
0330      STA $0222   ; DE V.B.I.
0340      PLA         ; RECUPERAR
0350      STA $D40E   ; INTERRUPCIONES
0360      CLI         ; TODAS!
0370      RTS         ; VOLVER
0380 ;
0390 ;INTERRUPCION
0400 ;
0410 VBI
0420 ;
0430      LDA $0231   ; NUESTRA LISTA
0440      CMP # >DL   ; ACTIVADA?
0450      BEQ DLOK    ; SI!
0460      LDA $0230
0470      STA $CB
0480      CLC
0490      ADC #$03
0500      STA DONDE   ; PARA SALTAR
0510      LDA $0231
0520      STA $CC
0530      ADC #$00
0540      STA DONDE+1 ; LISTA S.O.
0550      LDY #$00
0560 LOOP
0570      LDA ($CB),Y ; BUSCA FIN
```

```
0580      INY
0590      CMP #$41     ; SALTO?
0600      BNE LOOP     ; NO, BUSQUE
0610      LDA # <DL    ; APUNTE A
0620      STA ($CB),Y  ; NUESTRA
0630      STA $0230    ; LISTA DE
0640      INY          ; DESPLIEGUE
0650      LDA # >DL    ; AL FINAL
0660      STA ($CB),Y  ; DE CADA
0670      STA $0231    ; PANTALLA
0680 DLOK
0690      DEC SCROLL   ; PUNTERO
0700      BPL AUNNO    ; PIXELS
0710      LDA #$07     ; COMIENZE
0720      STA SCROLL   ; OTRO BYTE
0730      INC TEXT     ; CAMBIE EL
0740      BNE NOCARRY  ; PUNTERO DE
0750      INC TEXT+1   ; DESPLIEGUE
0760 NOCARRY
0770      LDA TEXT     ; FIN DEL
0780      CMP # <FINTEXT ; TEXTO?
0790      BNE AUNNO    ; AUN NO.
0800      LDA TEXT+1   ; REVISAR
0810      CMP # >FINTEXT ; DOS BYTES
0820      BNE AUNNO    ; FALTA!
0830      LDA # <DATA  ; RECOMIENZE
0840      STA TEXT     ; DESPLIEGUE
0850      LDA # >DATA  ; DESDE EL
0860      STA TEXT+1   ; PRINCIPIO
0870 AUNNO
0880      LDA SCROLL   ; MUEVA
0890      STA $D404    ; AHORA
0900      JMP $E45F    ; FIN RUTINA.
0910 ;
0920 ;
0930 ; DATOS
0940 ;
0950 SCROLL .BYTE 8   ; 8 PIXELS * BYTE
0960 ;
0970 DL
0980      .BYTE 112,112 ; 16 LINEAS
0990      .BYTE 86     ; LMS+GR.1+HSCROL
L
1000 TEXT
1010      .WORD DATA   ; DONDE ESTA
1020      .BYTE 1      ; SALTO
1030 DONDE ;           LISTA DESPLIEGUE
1040      .WORD $FFFF  ; SISTEMA OPERATI
VO
1050 DATA ;            TEXTO
1060      .WORD 0,0,0,0,0,0,0,0,0,0
1070      .SBYTE "revista turbo news"
1080      .SBYTE "1989   "
1090      .SBYTE "PEDRO PABLO "
1100      .SBYTE "CARABALL ALVAREZ  "
1110      *=  ORG+234 ;NO MAS DE $700
1120 FINTEXT
1130      .WORD 0,0,0,0,0,0,0,0,0,0
```

Curso Basic

```
10 REM ****************************
20 REM *  REVISTA TURBO NEWS (R)   *
30 REM *      MAURO PIERESSA       *
40 REM *      CURSO DE BASIC       *
50 REM *                           *
52 REM * OBJETIVO: Programa de fac-*
54 REM * turacion para demostra-   *
56 REM * cion de instrucciones.    *
60 REM ****************************
70 DIM FECHA$(30),NOMBRE$(20),RUT$(12)
,DOMICILIO$(30),COMUNA$(15),TELEFONO$(
10),GIRO$(30),DATOS$(2),DETALLE$(30)
80 DIM LINEA$(80)
90 LINEA$(1)=" ":LINEA$(80)=" ":LINEA$
(2)=LINEA$
92 REM ****************************
94 REM *      DATOS GENERALES      *
96 REM ****************************
100 ? "K     PROGRAMA DE FACTURACION":
? :?
110 TRAP 110:? "INGRESE FECHA: ";:INPU
T FECHA$
120 TRAP 120:? "INGRESE NOMBRE: ";:INP
UT NOMBRE$
130 TRAP 130:? "INGRESE RUT: ";:INPUT
RUT$
140 TRAP 140:? "INGRESE DOMICILIO: ";:
INPUT DOMICILIO$
150 TRAP 150:? "INGRESE COMUNA: ";:INP
UT COMUNA$
160 TRAP 160:? "INGRESE TELEFONO: ";:I
NPUT TELEFONO$
170 TRAP 170:? "INGRESE GIRO: ";:INPUT
 GIRO$:?
180 TRAP 40000:? "DATOS CORRECTOS(S/N)
: ";:INPUT DATOS$
190 IF DATOS$="N" THEN 100
200 LINEA$(50)=FECHA$:GOSUB 1000
210 LINEA$(1)=NOMBRE$:LINEA$(68)=RUT$:
GOSUB 1000
220 LINEA$(1)=DOMICILIO$:LINEA$(65)=CO
MUNA$:GOSUB 1000
230 LINEA$(1)=TELEFONO$:LINEA$(50)=GIR
O$:GOSUB 1000:LPRINT :LPRINT
240 SUBTOTAL=0
250 REM ****************************
260 REM *          DETALLE          *
270 REM ****************************
300 TRAP 200:? "INGRESE CANTIDAD (0 PA
RA TERMINAR ";:INPUT CANTIDAD
310 IF CANTIDAD=0 THEN 400
320 TRAP 220:? "INGRESE DETALLE: ";:IN
PUT DETALLE$
330 TRAP 230:? "INGRESE PRECIO UNITARI
```

```
0: ";:INPUT PRECIO
340 TOTAL=CANTIDAD*PRECIO:SUBTOTAL=SUB
TOTAL+TOTAL
350 LINEA$(1)=STR$(CANTIDAD):LINEA$(10
)=DETALLE$:LINEA$(55)=STR$(PRECIO):LIN
EA$(65)=STR$(TOTAL):GOSUB 1000
360 GOTO 300
370 REM ****************************
380 REM *          FINAL           *
390 REM ****************************
400 LINEA$(55)="SUBTOTAL":LINEA$(65)=S
TR$(SUBTOTAL):GOSUB 1000
410 LINEA$(55)="16% IVA":LINEA$(65)=ST
R$(SUBTOTAL*0.16):GOSUB 1000
420 LINEA$(55)="TOTAL":LINEA$(65)=STR$
(SUBTOTAL*1.16):GOSUB 1000
430 GOTO 100
1000 REM ***************************
1010 REM *         IMPRESION       *
1020 REM ***************************
1030 LPRINT LINEA$:LINEA$(1)=" ":LINEA
$(80)=" ":LINEA$(2)=LINEA$:RETURN
```

Desarrollando Hardware

```
100 REM ***************************
110 REM *   REVISTA TURBO NEWS    *
120 REM *                         *
130 REM * ARTICULO:               *
140 REM *  DESARROLLANDO HARDWARE *
150 REM *                         *
160 REM * POR:                    *
170 REM *     PEDRO P. CARABALL A.*
180 REM ***************************
190 REM
200 REM
210 REM SUBRUTINAS USR
220 REM
230 AB=ADR("PEDRO P. "):REM <> 0
240 AC=ADR("CARABALL "):REM = 0
250 RESTORE
260 FOR I=0 TO 8:READ J
270   POKE AB+I,J:POKE AC+I,J
280 NEXT I:POKE AC+6,208
290 DATA 104,173,0,211,41
300 DATA 48,240,249,96
310 AD=ADR("TURBO NEWS 1989")
320 FOR I=0 TO 14
330   READ J:POKE AD+I,J
340 NEXT I
350 DATA 104,133,213,173,0,211,41
360 DATA 48,74,74,74,74,133,212,96
370 REM
380 REM DIBUJA PANTALLA
390 REM
400 GRAPHICS 0
410 POKE 710,0:POKE 752,1
420 DL=PEEK(560)+PEEK(561)*256
430 POKE DL+3,70:POKE DL+6,7
440 ? "♥♥♥♥♥COMBATE♥♥♥♥♥♥♥♥♥♥♥♥♥♥naval
"
450 DIM A$(18):? :?
460 A$=" 0123456789ABCDEF "
470 ? A$;A$
480 A$(3,18)="................"
490 FOR I=1 TO 15:POSITION 2,I+3
500   N=ASC(A$(2)):IF N=57 THEN N=64
510   N=N+1:A$(2)=CHR$(N):A$(18)=" "
520   ? A$;A$:NEXT I
530 REM
540 REM DIBUJA BARCOS
550 REM
560 P=PEEK(88)+PEEK(89)*256
570 D=160:T=15
580 FOR L=1 TO 5:C=(L=4)*2+(L=5)*4
590   C=L-C:FOR J=1 TO C
600     X=INT(RND(0)*T)
610     Y=INT(RND(0)*T)
620     IF X+L>18 THEN 600
630     N=D+X+Y*40+P
640     FOR I=1 TO L:IF PEEK(N+I)<>14
THEN POP :GOTO 600
650     NEXT I
660     FOR I=1 TO L:POKE N+I,128
670     NEXT I:NEXT J:NEXT L
680 REM
690 REM JOYSTICK 2 ENTRADA/SALIDA
700 REM
710 U=54016:Z=0:ZZ=Z
720 POKE 54018,48:POKE U,192
730 POKE 54018,60:POKE U,192
740 REM
750 REM QUIEN JUEGA PRIMERO?
760 REM
770 E=USR(AB):E=USR(AD)
780 REM
790 REM E=1 > TU PRIMERO
800 REM
810 IF E=1 THEN POKE U,128:C=USR(AC):G
OTO 1670
820 REM
830 REM E=3 > COMENZANDO
840 REM
850 IF E=3 THEN POKE U,64
860 E=USR(AD)
870 IF E=1 THEN 960
880 REM
890 REM E=2 > YO PRIMERO
900 REM
910 IF E=2 THEN POKE U,0:GOTO 1880
920 GOTO 860
930 REM
940 REM DOS AL MISMO TIEMPO NO!
950 REM
```

```
960 POKE U,192
970 POKE 540,RND(0)*100
980 IF PEEK(540) THEN 980
990 GOTO 770
1000 REM
1010 REM FALLASTE EL DISPARO!
1020 REM
1030 POKE U,255:POKE P+X+Y,56
1040 C=USR(AC):POKE U,0
1050 REM
1060 REM MOVIMIENTO DEL JOYSTICK
1070 REM
1080 X=22:Y=160:S=PEEK(P+X+Y)
1090 J=PEEK(632):POKE P+X+Y,S
1100 SOUND 0,Y/50+X,10,2
1110 Y=Y+(J=14)*-40+(J=13)*40
1120 Y=Y+(Y<160)*40+(Y>720)*-40
1130 X=X+(J=11)*-1+(J=7)
1140 X=X+(X<22)+(X>36)*-1
1150 S=PEEK(P+X+Y):POKE P+X+Y,142
1160 SOUND 0,0,0,0:POKE 77,0
1170 IF S<>14 THEN 1090
1180 IF PEEK(644) THEN 1090
1190 REM
1200 REM TRANSFORMAR COORDENADAS
1210 REM
1220 Y1=(Y-160)/40:X1=X-22
1230 Y1=Y1*16:B=X1+Y1
1240 REM
1250 REM TRANSMITIR
1260 REM
1270 FOR I=7 TO 0 STEP -1
1280   POKE U,0:K=2^I:V=192
1290   IF (B<K) THEN V=128:K=0
1300   C=USR(AC):POKE U,V:B=B-K
1310   IF USR(AD)<>1 THEN 1310
1320 NEXT I
1330 REM
1340 REM RECIBIR AGUA O ACIERTO
1350 REM
1360 IF USR(AD)=3 THEN 1600
1370 IF USR(AD)<>2 THEN 1360
1380 REM
1390 REM ACERTE!!!
1400 REM
1410 POKE P+X+Y,184:Z=Z+1
1420 FOR A=15 TO 0 STEP -0.1
1430   SOUND 0,60,0,A
1440 NEXT A
1450 IF Z<27 THEN 1670
1460 REM
1470 REM GANE ESTE JUEGO
1480 REM
1490 POKE U,0:? :? ">>GANASTE"
1500 FOR A=255 TO 0 STEP -2
1510   POKE 712,A
1520   SOUND 0,A,10,A/25
1530   SOUND 1,A/2,10,A/25
1540 NEXT A
1550 IF PEEK(644) THEN 1550
1560 RUN
1570 REM
1580 REM FALLE!
1590 REM
1600 POKE P+X+Y,56
1610 FOR A=15 TO 0 STEP -0.15
1620   SOUND 0,2,8,A
1630 NEXT A
1640 REM
1650 REM RECIBIR
1660 REM
1670 POKE U,0:B=0
1680 FOR I=7 TO 0 STEP -1
1690   C=USR(AC):POKE U,0
1700   C=USR(AB):E=USR(AD)
1710   POKE U,64
1720   E=E-2:B=B+2^I*E
1730 NEXT I
1740 REM
1750 REM RECUPERAR COORDENADAS
1760 REM
1770 Y=INT(B/16):X=B-Y*16
1780 Y=Y*40+160:X=X+4
1790 REM
1800 REM ACERTO?
1810 REM
1820 IF PEEK(P+X+Y)<>128 THEN 1030
1830 REM
1840 REM UGH! ACERTASTE
1850 REM
1860 POKE U,128:POKE P+X+Y,184
1870 C=USR(AC)
1880 ZZ=ZZ+1:IF ZZ<27 THEN 1080
1890 REM
1900 REM PERDI, QUE PENITA!
1910 REM
1920 POKE U,255:? :? ">>PERDISTE"
1930 FOR I=0 TO 3
1940   FOR A=15 TO 0 STEP -0.5
1950     SOUND 0,160+I*30,10,A
1960   NEXT A:NEXT I
1970 IF PEEK(644) THEN 1970
1980 RUN
```

Mapa de memoria

```
10 REM ****************************
20 REM *     REVISTA TURBO NEWS   *
30 REM *                          *
40 REM *       MAPA DE MEMORIA    *
50 REM *                          *
60 REM *       USOS PRACTICOS     *
70 REM *                          *
80 REM *   PEDRO P. CARABALL A.   *
```

```
90 REM ******************************
95 TRAP 1330:POKE 710,0
100 REM
110 REM DESCONECCION TECLA BREAK
120 REM
130 REM B=PEEK(16)-128:IF B>0 THEN POK
E 16,B:POKE 53774,B
200 REM
210 REM INICIALIZACION DE RELOJ
220 REM
230 ? "INGRESE HORAS ";:INPUT H
240 ? "INGRESE MINUTOS ";:INPUT M
250 ? "INGRESE SEGUNDOS ";:INPUT S
260 T=H*216000+M*3600+S*60
270 T1=INT(T/65536):T=T-T1*65536
280 T2=INT(T/256):T=T-T2*256
290 POKE 18,T1:POKE 19,T2:POKE 20,T
300 REM
310 REM CONSULTA DE RELOJ
320 REM
330 T=PEEK(20)+PEEK(19)*256+PEEK(18)*6
5536
340 IF T<5184000 THEN 380
350 POKE 18,PEEK(18)-79
360 POKE 19,PEEK(19)-26
370 GOTO 330
380 H=INT(T/216000):T=T-H*216000
390 M=INT(T/3600):T=T-M*3600
400 S=INT(T/60)
410 ? "HORA >";H;":";M;":";S;"<"
500 REM
510 REM SONIDO ENTRADA SALIDA
520 REM
530 ? "SONIDO DE ENTRADA/SALIDA"
540 ? "0 PARA DESACTIVAR"
550 ? "1 PARA ACTIVAR"
560 ? "OPCION ";:INPUT S
570 POKE 65,S
600 REM
610 REM ROTACION DE COLORES
620 REM
630 POKE 77,128
640 FOR I=1 TO 500
650 ? "ROTANDO COLORES ",
660 NEXT I
670 POKE 77,0
680 ? :? "FIN ROTACION"
700 REM
710 REM USO DE MARGENES
720 REM
730 FOR I=0 TO 20
740 POKE 82,I
750 POKE 83,39-I
760 ? :? "LINEA DEMOSTRACION - CUARENT
A CARACTERES"
770 NEXT I
780 POKE 82,2
790 POKE 83,39
800 REM
810 REM POSICION DE CURSOR
820 REM
830 ? CHR$(125)
840 FOR I=0 TO 200
850 X=RND(0)*23:POKE 84,X
860 Y=RND(0)*39:POKE 85,Y
870 ? "."
880 SOUND 0,RND(0)*40+20,14,10-I/20
890 NEXT I
900 REM
910 REM USO DEL PUNTERO DE PANTALLA
920 REM
930 P=PEEK(88)+PEEK(89)*256
940 LIST 800,9090
950 FOR Y=0 TO 23
960 FOR X=0 TO 19
970 N=P+Y*40+X
980 N1=(N+39-X-X)
990 B=PEEK(N)
1000 POKE N,PEEK(N1)
1010 POKE N1,B
1020 NEXT X
1030 NEXT Y
1100 REM
1110 REM VARIABLES EN USO
1120 REM
1130 INI=PEEK(130)+PEEK(131)*256
1140 FIN=PEEK(132)+PEEK(133)*256-1
1150 ? :X=0:FOR V=INI TO FIN
1160 N=(PEEK(V)>127)*128:X=X+(N>0)
1170 ? CHR$(PEEK(V)+N);CHR$(27+N);
1180 NEXT V
1190 ? "NUMERO DE VARIABLES EN USO =";
X
1200 REM
1210 REM LINEA ACTUAL
1220 REM
1230 ? "TRABAJANDO EN LINEA > ";PEEK(1
60)+PEEK(161)*256
1240 END
1300 REM
1310 REM CONTROL DE DETENCION Y ERROR
1320 REM
1330 LIN=PEEK(186)+PEEK(187)*256
1340 TRAP 1330:? CHR$(253):? :?
1350 ? "ERROR NUMERO >";PEEK(195);
1360 ? "< EN LINEA NUMERO ";LIN
1370 ? :? "CONTINUO CON LA EJECUCION?"
1380 ? "0- DETENER"
1390 ? "1- CONTINUAR"
1400 ? :? "SU OPCION ";:INPUT X
1410 IF X THEN GOTO LIN
1420 END
```

La nueva generación de Software para Computadores Atari

ADQUIERALOS EN LOS SIGUIENTES PUNTOS DE VENTAS

• **ANTOFAGASTA:** COOPERCARAB KW VIDEO LA ESPAÑOLA • **VIÑA DEL MAR:** FALABELLA VIÑA INSIS MPR COMPUTACION • **VALPARAISO:** COMPUTRONIC • **SANTIAGO:** AUDIO BICICLETA INTERNAC CASA ROYAL CENTRO ATARI COMERCIAL ESTADO COMPUMANQUE COMPUCENTER FALABELLA AHUMADA FALABELLA P. ARAUCO IMACO INFOGROUP PC STORE PETERSEN ROLEC SUPERMERCADOS UNIMARC TASCO VIDEO CLUB INTERNACIONAL • **RANCAGUA:** CASA ZUÑIGA • **CURICO:** MULTIHOGAR • **TALCA:** LIBRERIA "EL AHORRO" MULTICENTRO VIDEO CLUB CASSAL • **CHILLAN:** CASA EDISON • **CONCEPCION:** COOPERCARAB DISMAR DISMAR 2 EQUS PHANTER RAPSODIA SESCO • **LOS ANGELES:** DISTRIBUIDORA MERINO • **ANGOL:** SCORPIO • **VICTORIA:** CASA SIGMUND • **TEMUCO:** COMERCIAL MANQUEHUE ESTABLECIMIENTOS GEJMAN FALABELLA • **PUCON:** ELTIT • **VILLARRICA:** JOYERIA KETTERER • **VALDIVIA:** ELECTROMUSICA • **LA UNION:** IMPORTADORA COSMOS • **OSORNO:** CASA REAL FOTO EXPRESS • **PUERTO VARAS:** ELECTRO HORN • **PUERTO MONTT:** COMERCIAL MANQUEHUE DIMARSA • **COYHAIQUE:** FACI HOGAR • **PUNTA ARENAS:** BALFER LTDA.

En pocos segundos gane mucho tiempo

NUEVA

DISKETTERA ATARI XFF-551

La diskettera Atari XFF-551, impone una nueva velocidad de trabajo al computador ATARI XL/XE Sólo unos pocos segundos bastan, para que la unidad de diskettes magnéticos de 5¼ pulgadas XFF cargue completamente un programa en la memoria de su computador Sin demoras, sin esperas, sin errores. Para un veloz acceso a la información que necesita, incorpórela rápidamente a su computador y pase adelante a toda prisa.

EXIJA ESTE SELLO
DEL REPRESENTANTE EXCLUSIVO
COELSA COMPUTACION
GARANTIA TOTAL

ATARI®
COMPUTADORES

COELSA
COMPUTACION
Sinónimo de garantía y servicio.

Ocho de cada diez computadores para el hogar, son ATARI